

**FINAL REPORT ON THE
DIGITAL AUDIO CAPTURE AND
IDENTIFICATION SYSTEM (DACIS)**

FOR THE REMOTE ENVIRONMENTAL SENSING
PROGRAM

AUGUST 1998

Protagoras N. Cutchis
Space Department
Johns Hopkins University Applied Physics laboratory

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

REPORT DOCUMENTATION PAGEForm Approved
OMB No. 074-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503

1. AGENCY USE ONLY (Leave blank)**2. REPORT DATE**
August 1998**3. REPORT TYPE AND DATES COVERED**
Final Report**4. TITLE AND SUBTITLE**

Final Report on the Digital Audio Capture and Identification System (DACIS).
SERDP: Advanced Biotelemetry for Resource Management on Military Lands (CS-759) , Part II

5. FUNDING NUMBERS

N/A

6. AUTHOR(S)

Protagoras N. Cutchis

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Johns Hopkins University Applied Physics Laboratory

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

SERDP
901 North Stuart St. Suite 303
Arlington, VA 22203

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

N/A

11. SUPPLEMENTARY NOTES

No copyright is asserted in the United States under Title 17, U.S. code. The U.S. Government has a royalty-free license to exercise all rights under the copyright claimed herein for Government purposes. All other rights are reserved by the copyright owner.

12a. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release: distribution is unlimited.

12b. DISTRIBUTION CODE

A

13. ABSTRACT (Maximum 200 Words)

The Digital Audio Capture and Recognition System (DACIS) is designed to be an animal mounted (eventually bird borne) system which records and analyzes animal noises automatically. While the DACIS is fairly complex, it is actually an advanced acoustic sensor from a system point of view. The goal of the prototype system is to be able to identify with a high degree of reliability 4 animal calls from a single or multiple species. A very limited number of tests were performed on the DACIS with four types of wolf calls that were named growls, howls, whines and barks. A threshold value of 17 was found to be optimal yielding no false positives and 25% false negatives I early testing. Later tests showed up to 50% false negatives but still no false positives with a threshold of 20 or lower.

14. SUBJECT TERMS

SERDP, DACIS, biotelemetry, biodiversity, natural resource management, wildlife

15. NUMBER OF PAGES

140

16. PRICE CODE

N/A

17. SECURITY CLASSIFICATION OF REPORT

unclass

18. SECURITY CLASSIFICATION OF THIS PAGE

unclass

19. SECURITY CLASSIFICATION OF ABSTRACT

unclass

20. LIMITATION OF ABSTRACT

UL

Applied Physics Laboratory

11100 Johns Hopkins Road
Laurel MD 20723-6099
240-228-5000 / Washington
443-778-5000 / Baltimore

Please refer to:
CMI-98-040

31 August 1998

***Institute for Advanced Science
and Technology in Medicine***

To: WS Seegar

From: PN Cutchis, M. D.

Subject: Final Report on the DACIS (Digital Audio Capture and Identification System)

Enclosures:

- (1) Schematics of DACIS Processor Board
- (2) Assembly Language Software for DACIS unit
- (3) BASIC Software for PC Communication/Test Software
- (4) Raw and analyzed data from DACIS for wolf calls
- (5) Article from September 1997 NASA conference on DACIS

INTRODUCTION AND GENERAL DESCRIPTION OF THE DIGITAL AUDIO CAPTURE AND RECOGNITION SYSTEM (DACIS)

The DACIS system is designed to be an animal mounted (eventually bird borne) system which records and analyzes animal noises automatically. While the DACIS is fairly complex, it is actually an advanced acoustic sensor from a system point of view. The goal of the prototype system described here is to be able to identify with a high degree of reliability 4 animal calls from a single or multiple species. The system is based on an 8-bit MC68HC811 microprocessor that contains EEPROM, an 8-bit A/D converter and a serial interface, all of which are used in the DACIS design. The prototype board is 1.95" square and weighs 18.2 grams. This version incorporates a socket, which increase the size and weight of the board. The final design, which has been laid out, is 1.85" square and should weigh less than 15 grams.

The present design captures 5 seconds of audio at 6,000 samples per second to yield a theoretical high frequency capture of 3 kHz. The system then performs 10 digital filters on each of ten 0.5 second time epochs to yield a matrix of 100 time/frequency parameters. These parameters are then normalized to account for volume level differences and then compared to a stored table using a pattern-matching algorithm.

The DACIS unit, while in the listen mode, only has the microphone, 2-stage amplifier, and threshold comparator circuit operating. In this mode, the unit consumes only 900-1300 μ A of current at 5 VDC. If the volume level exceeds a preset threshold (adjustable on the board), then the

processor is awoken from its STOP (low power) mode and starts the digital 5 second recording. While recording and performing the analysis, the DACIS system consumes approximately 16 mA and this analysis takes 17 minutes per animal call.

SOFTWARE DESCRIPTION

The assembly language software for the DACIS includes subroutines that perform the following major tasks:

- 1) Interrupt trigger of processor
- 2) Digital 5 second recording
- 3) 10 frequency band digital band-pass filters
- 4) Subroutine for normalization of filter outputs
- 5) Pattern matching against four stored templates
- 6) Communication with Dahlgren furnished ARGOS PTT host microprocessor

The software for the DACIS system was written entirely in assembly language and the test software for the host PC was written in BASIC. The software initializes memory locations and other parameters and then executes the STOP command that halts all microprocessor functions including the oscillator for maximum power conservation. The IRQ (Interrupt ReQuest) line, which is pulled low by the comparator when the audio volume exceeds a preset threshold, awakens the processor from the stopped state. The processor then records 5 seconds of audio at 6,000 samples per second. Then, the software, in the routine DIGIFILT (for digital filter) performs 10 FIR (Finite Impulse Response) filters at frequencies from 250 Hz to 2500 Hz in 250 Hz increments. The filters are 23 stage. This number was determined from simulations run on a PC to obtain suitable fall off without too much filter overlap. The 10 filters are run on each of 10 time epochs which are approximately 0.5 seconds in length (the exact length is 0.512 seconds to ease software design).

After completing the digital filters, the 100 parameters obtained are normalized to compensate for differences in volume level during recording. The next routine is the pattern match routine which performs the least squares fit to the 100 template parameters for each of the four stored templates. The formula used is:

$$x = \sqrt{\frac{\sum (a_{ij} - c_{ij})^2}{100}}$$

where a_{ij} and c_{ij} are the filter outputs and template parameters respectively. The last routine that was going to be implemented was simple threshold detection for positive identification of the calls. It was decided to not implement that routine (although it is coded into the DACIS processor) and instead to transfer all four correlation values from the above equation for each call to the Dahlgren unit.

There can be timing problems with missed bytes if the PC utilized is a very new fast processor. Adjusting timing loops in the PC code can compensate for this.

HARDWARE DESCRIPTION

The design of the hardware is complete. The MC68HC11 microprocessor contains internal RAM (Random Access memory), EEPROM (Electrically Erasable Read Only Memory), a UART (Universal Asynchronous Receiver Transmitter) and A/D (Analog to Digital) converter subsystems on-chip which are used in this design.

A two-stage amplifier built from an LM358AM dual operational amplifier amplifies the microphone's signal. Then, the signal is fed into an LM339 voltage comparator. When the sound level exceeds a threshold, which is set by an on board trimmer potentiometer, the IRQ line of the processor is brought low and the processor awakens from its low power "sleep" state and starts recording. The memory is a 128K byte by 8 static RAM of which all 128K can be addressed in the present hardware. However, the present software only requires about 32K.

Two versions of the DACIS processor were laid out and 2 prototypes of one version were fabricated. One version has no socket for the microprocessor and the second version, which is slightly larger, has a socket. The sizes are 1.85" square and 1.95" square respectively. The version with the socket version has been fabricated since this version allows easy removal of the microprocessor for software changes.

POWER SUPPLY REQUIREMENTS

The DACIS board requires regulated 5 VDC $\pm 500\text{mV}$ as there is no on board voltage regulation. The current consumption varies somewhat from unit to unit but is between 900 μA and 1250 μA during the listen period and 15 mA and 34 mA during the record and signal analysis periods. The DACIS system, other than audio "awakening" has no internal method to completely shut itself off. It shuts down all of its own internal timers while in the "stop" mode to save power and therefore has no reference of how much time has passed. Therefore, in the Dahlgren application, the Dahlgren unit, which does have a real-time clock, will be responsible for time keeping. The maximum number of calls to be analyzed over any set time period can be programmed by having the Dahlgren processor shut down power to the entire DACIS unit until further analyses are desired.

DAHLGREN INTERFACE

The details of the Dahlgren interface were not confirmed until August of 1998. The interface functions as follows. There are 2 serial lines (one in each direction), 0-5 volts at 9,600 baud, 8 bits, 1 stop bit and no parity between the two processors. The DACIS will keep the transmit line at high impedance until it is given permission to transmit data (see below). There are 2

additional handshaking lines, again one in each direction between the two processors. After analyzing a call, the DACIS will raise its line high to signal the Dahlgren processor that it has data available for transfer. When it is ready, the Dahlgren processor will raise its line to the DACIS high signaling that it is prepared for data transfer. The DACIS will immediately enable the transmitter and await an ASCII "S" from the Dahlgren unit over the serial link. The DACIS will wait essentially forever in an infinite loop until it sees this "S" or an "R" for resend. The DACIS will then transfer 4 bytes that correspond to the four template matches. After transferring the data once, the DACIS will wait for about 0.5 seconds to see if an "R" or "S" is sent to request that the data be sent again. If the DACIS does not receive an "R" or "S" within the 0.5 seconds it will reset and wait to analyze the next call. If it does it will wait again for another 0.5 seconds. The unit can be kept in this mode, resending the same data forever if requests are continually sent within 0.5 second windows.

The last portion of the Dahlgren interface is that the Dahlgren processor will have control over the DACIS power lines so that if too many analyses are being conducted, the Dahlgren unit can shut down the DACIS board temporarily to save system battery power.

The wiring connections on the ribbon cable to the DACIS board are as follows:

BLUE:	Ground
GREEN:	Dahlgren to DACIS control line
YELLOW:	DACIS to Dahlgren control line
ORANGE:	Serial data output
RED:	Serial data input
BROWN:	+5VDC

TRIGGER SOUND LEVEL ADJUSTMENTS

The gain of the two stage audio amplifier is fixed at a level that was found to be suitable for most anticipated wildlife environments. The goal was to obtain reasonable gain without the amplifier saturating. However, the trigger threshold is adjustable by adjusting a small trimmer potentiometer on the back (opposite side from the processor) of the board. The setting of this potentiometer is quite "touchy". All of the trimmers were set at APL during testing to a level that seemed appropriate. However, without actual field data to indicate most likely volume levels for a specific species, the level set at APL may need further adjustment. If the level is set to be too sensitive, the unit will trigger too frequently on spurious sounds. Therefore, it is recommended that the unit be set slightly to the less sensitive side of what may be perceived as optimum. The setting found at APL to be best yields a voltage of 1.5 volts at pin 4 of the LM339 comparator.

CMI-98-040

Page 5

TEST RESULTS

A very limited number of tests were performed on the DACIS with four types of wolf calls that were named growls, howls, whines and barks. The howls and barks frequently blend together. This is one of several factors which indicate that from an acoustic standpoint, wolves appear not have been an "easy" first target animal for the DACIS system. Wolves were chosen because of the availability of acoustic data and because they are large enough to carry a large collar with the Dahlgren Argos/GPS unit.

The data were first analyzed on a PC in Excel with digital filter outputs transferred across a serial link from the DACIS. This was done so that actual wolf calls, as recorded through the DACIS microphone and recorded by its processor could be used. The threshold value, which can now be selected at the analysis station, can be moved up or down so as to reduce false positives or false negatives or the total error rate. As the spreadsheets show in Enclosure 4, a threshold value of 17 was found to be optimal yielding no false positives and 25% false negatives in early testing. Later tests showed up to 50% false negatives but still no false positives with a threshold of 20 or lower.

An additional analog data tape received in August 1998 provided few if any wolf calls of quality worthy of conducting additional testing. Therefore, after discussions with the sponsor, it was decided not to conduct further analyses. It was agreed upon to instead load the existing template parameters into the DACIS and to complete the software for the Dahlgren interface.



PN Cutchis, M.D.

PNC:cco

CMI-98-040

Page 6

Internal Distribution:

PN Cutchis (w/enclosures)

B Henke (w/enclosures)



JC Murphy (w/enclosures)

WS Seegar (w/enclosures)

CMI Files (w/enclosures)

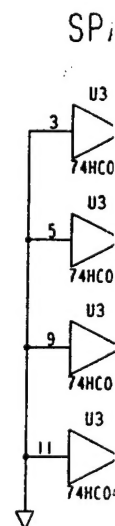
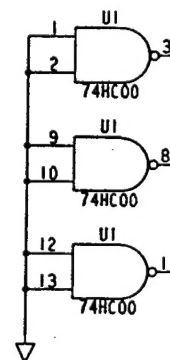
Enclosure 1
Schematics of DACIS Processor Board

NOTES:

- WHERE APPLICABLE, UNLESS OTHERWISE SPECIFIED:
ALL RESISTANCES ARE IN OHMS
ALL CAPACITANCES ARE IN FARADS
ALL VOLTAGES ARE DC
THE SYMBOL  REPRESENTS A SPACE
 () SHEET TO SHEET REFERENCES
- THE FOLLOWING ARE THE LAST REF DESIGNATIONS USED:
C13, MK1, R18, U8, Y1
- THE FOLLOWING ARE UNUSED REF DESIGNATIONS:
NONE
- UNLESS SHOWN ON THE ACTUAL CIRCUITRY, ALL POWER CONNECTIONS FOR INTEGRATED CIRCUITS WILL BE REPRESENTED ON THE "IC POWER/GROUND CHART"

IC POWER/GROUND CHART

REF. NO.	PART NO.	GENERIC NO.	
14	U1	74HC00	7
14	U3	74HC04	7
20	U8	74HC373	10
	U5	LM358A	16
	U6	LM339	
	U7	MC68HC81F2CFN	
	U4	S-8054HN-CR-Y	
	U2	SRW20100LMT70	



5301-1851

2	1	SWED	SW SWED
6	6	017	01 SWED

QUESTION 10				
Q#	Q# 10a1	QUESTION 10a2	Q# 10a3	QUESTION 10a4

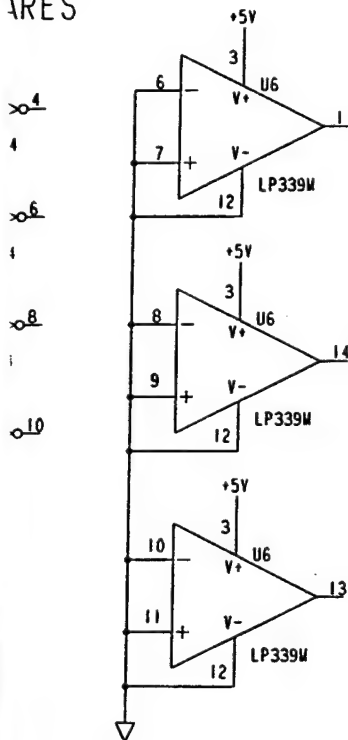
E

6

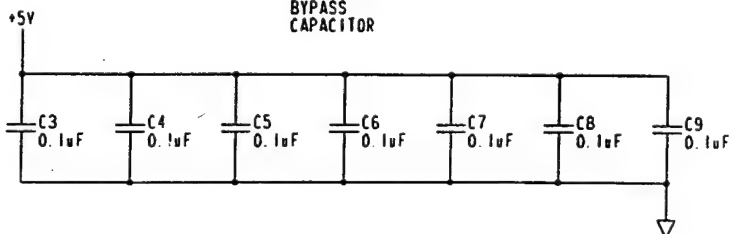
8

10:13:00

ARES



ALL ICs
EXCEPT U4
GET 0.1 μ Fd
BYPASS
CAPACITOR



BASIC										
QTY / UNIT	UNIT NO.	DATE	NAME OF SERVICE PROVIDER	CATEGORY OF THE	ANALYTICAL OR DESCRIPTION	DURABLE IN USE	ORIGINALS, REPRODUCED MATERIALS, SPECIFICATIONS FOR	EASE CASE	BY	

[illegible]

SCHEMATIC
TRIGGERED BIRD #2
AUDIO CAPTURE SYSTEM
WITH 128K MEMORY

88898

D

5301-1851

10

MENTOR GRAPHICS V8.2

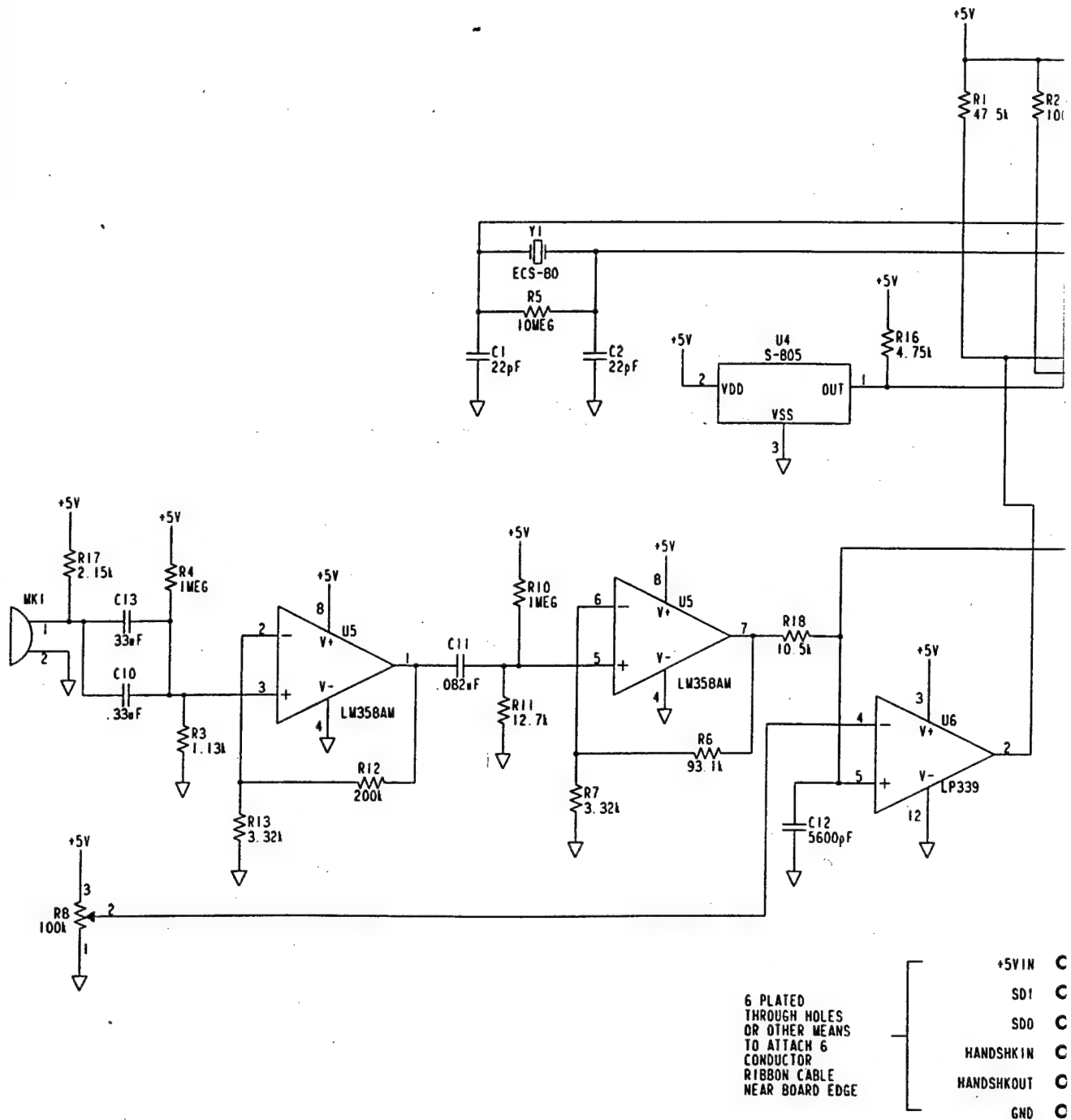
5301-1850

D

C

B

A



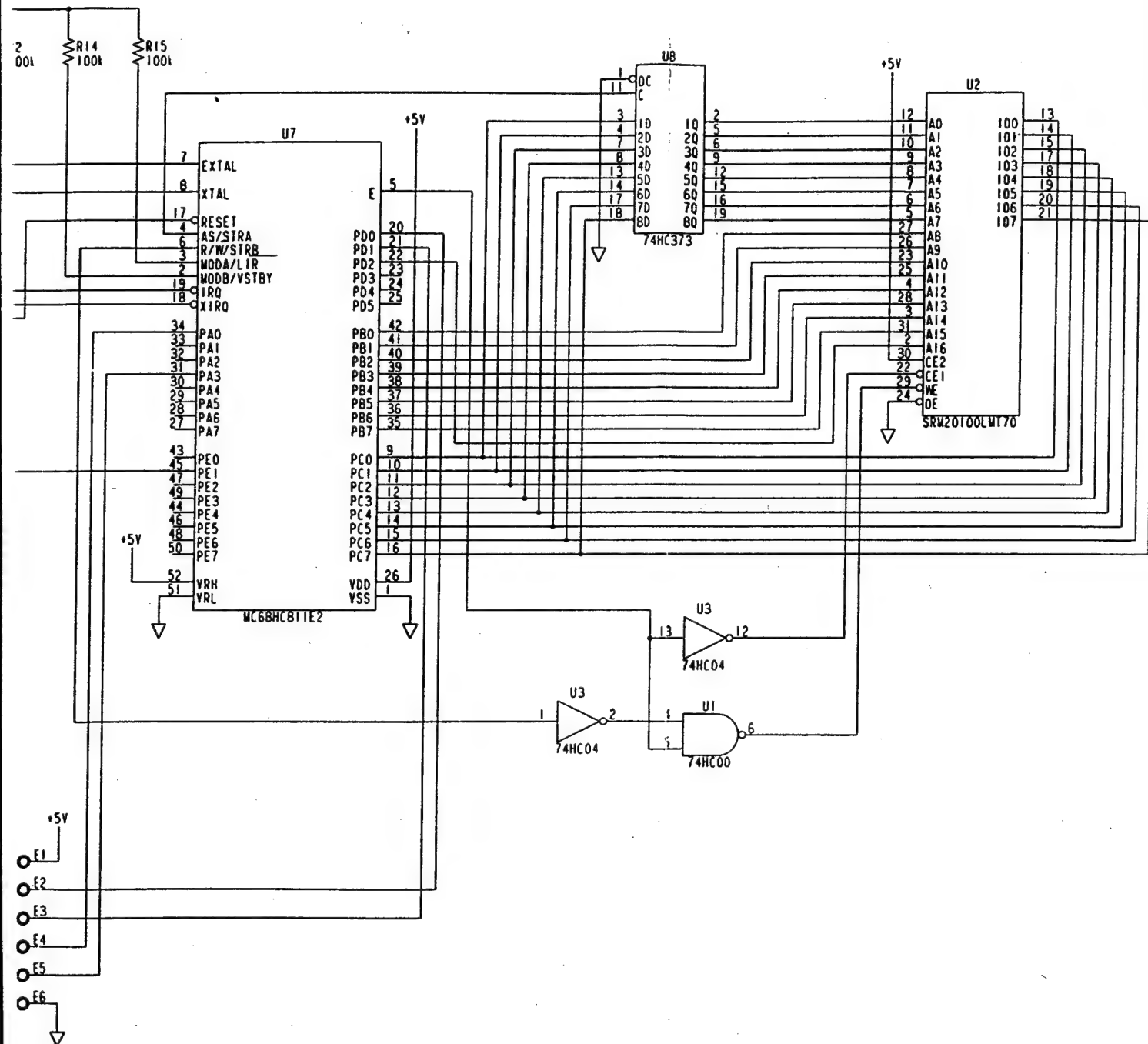
NOT ARCHIVED

7

6

5

REVISIONS				
REV	BY & DATE	DESCRIPTION	CHECK & DATE	APPROVED & DATE



88898	D	5301-1851	b
SCALE	NONE	NO OF SCALE POINT	2
MENTOR GRAPHICS V8.2	5301-1850		

NAME: 10-15-97

Enclosure 2
Assembly Language Software
For DACIS Processor

```
1 ; WOLF-MOUNTED DIGITAL AUDIO CAPTURE AND IDENTIFICATION
2 ; SYSTEM (DACIS) FOR THE REST PROGRAM
3 ; THIS PROGRAM IS DESIGNED FOR THE DAHLGREN INTERFACE
4 ; **** ALL 10 DIGITAL FREQUENCY FILTERS WITH 10 TIME EPOCHS ***
5 ; WITH AUDIO CAPTURED THROUGH THE MICROPHONE
6 ;
7 ; VERSION: C:\BIRD97\WOLF987.ASM
8 ;
9 ; THIS VERSION UTILIZES A 128 KBYTE MEMORY (SRM20100LMT)
10 ; & IMPLEMENTS THE LP339 COMPARATOR AUDIO THRESHOLD TRIGGER
11 ; CREATED: THURSDAY 8/27/98 2:25 PM
12 ; LAST REVISION: THURSDAY 8/27/98 2:50 PM
13 ; 1) CHANGED FROM IRQ TO XIRQ TO ALLOW STOP MODE USE 4/8/96
14 ; 2) CHANGED TO OPERATE ON 4.9152 MHZ XTAL (BAUD=001000001)
15 ; 3) COMPLETED ASCII SUBROUTINE 7/18/96
16 ; 4) REMAPPED INTERNAL ROM & RAM TO 8K AND 9K BOUNDARIES
17 ; 5) CHANGED MEAN TO 50 TO ALLOW 5:1 DYNAMIC RANGE IN FILTER BANDS
18 ; 6) REMOVED DATA TRANSFER CODE FROM DIGIFILT FOR FILTER OUTPUT 8/19/98
19 ; 7) REMOVED DATA TRANSFER CODE FROM NORMALIZE 8/19/98
20 ; 8) ADDED HSN AND HSOUT FOR DATA TRANSFER 8/19/98 (NO PA0 INTERRUPT)
21 ; 9) DUMP ALL 8 BYTES OF MATCH DATA FOR TEST ONLY: ADDED 8/25/98
22 ; LAST REVIEWED: THURSDAY 8/27/98
23
24 DEFSEG DIG, START=0F800H
25 SEG DIG
26 ;ROM ADDRESSES (REGISTERS ETC.)
27 INIT EQU $103D
28 PORTAADR EQU $8000
29 PORTCADR EQU PORTAADR+3
30 PORTBADR EQU PORTCADR+1
31 PORTCDDR EQU PORTBADR+3
32 PORTDADR EQU PORTCDDR+1
33 PORTDDDR EQU PORTDADR+1
34 TCNT EQU PORTDDDR+5
35 TOC1 EQU TCNT+8
36 TOC2 EQU TOC1+2
37 TCTL2 EQU TOC2+9
38 TMSK1 EQU TCTL2+1
39 TFLG1 EQU TMSK1+1
```

```
=103D
=8000
=8003
=8004
=8007
=8008
=8009
=800E
=8016
=8018
=8021
=8022
=8023
```

=8024	40 TMSK2	EQU	TFLG1+1			
=8026	41 PACTL	EQU	TMSK2+2			
=802B	42 BAUD	EQU	PACTL+5			
=802C	43 SCCR1	EQU	BAUD+1			
=802D	44 SCCR2	EQU	SCCR1+1			
=802E	45 SCSR	EQU	SCCR2+1			
=802F	46 SCDR	EQU	SCSR+1			
=8030	47 ADCTL	EQU	SCDR+1			
=8031	48 ADR1	EQU	ADCTL+1			
=8032	49 ADR2	EQU	ADR1+1			
=8033	50 ADR3	EQU	ADR2+1			
=8034	51 ADR4	EQU	ADR3+1			
=8039	52 OPTION	EQU	ADR4+5			
53 ;-----						
54 ;RAM ADDRESSES						
55 ;VARIABLE						
=9000	56 COUNT	REL. EQU	ADD. \$9000	ABS. ADD. ;0	FUNCTION	
=9001	57 TEMP1	EQU	COUNT+1	;1		
=9002	58 TEMP2	EQU	TEMP1+1	;2		
=9003	59 TEMP3	EQU	TEMP2+1	;3		
=9004	60 TIMINT	EQU	TEMP3+1	;4&5		SAMPLE TIME INT IN uS
=9006	61 NUMINT	EQU	TIMINT+2	;6		NUMBER OF PLAYBACK REPEATS
=9007	62 MEMPAGE	EQU	NUMINT+1	;7		MEMPAGE=0 OR 1
=9008	63 ENDDATA	EQU	MEMPAGE+1	;8 & 9		ADDRESS FOR END OF DATA
=900A	64 EPOCH	EQU	ENDDATA+2	;10		ADDRESS FOR EPOCH # (1-10)
=900B	65 INSTRUCT	EQU	EPOCH+1	;11		SINGLE BYTE INSTRUCTION
=900C	66 FILPAR	EQU	INSTRUCT+1	;12&13		PARAMETER INDEX
=900E	67 AVERAGE	EQU	FILPAR+2	;14&15		AVERAGE VALUE OF FILTER OUT
=9010	68 FILTAMP	EQU	AVERAGE+2	;16-115		100 FILTER OUTPUTS
=9074	69 NUMCALL1	EQU	FILTAMP+100	;116		NUMBER OF TYPE 1 CALLS
=9075	70 NUMCALL2	EQU	NUMCALL1+1	;117		NUMBER OF TYPE 2 CALLS
=9076	71 NUMCALL3	EQU	NUMCALL2+1	;118		NUMBER OF TYPE 3 CALLS
=9077	72 NUMCALL4	EQU	NUMCALL3+1	;119		NUMBER OF TYPE 4 CALLS
=9078	73 FREQ	EQU	NUMCALL4+1	;120		FILTER BEING USED (0-9)
=9079	74 FILTSTART	EQU	FREQ+1	;121-122		START ADDRESS OF FILTER
=907B	75 SIGN	EQU	FILTSTART+2	;123		BIT7=SIGN OF MULT RESULT
=907C	76 MEGAADDL	EQU	SIGN+1	;124-125		MEGA ADDITION REGIS-LOW
=907E	77 MEGAADDH	EQU	MEGAADDL+2	;126-127		MEGA ADDITION REGIS-HIGH
=9080	78 PCLOADFLG	EQU	MEGAADDH+2	;128		PC LOADING FLAG

```

=9081      79 TESTADDRS EQU PCLOADFLG+1 ;129&130      TEST ADDRESS FOR PC DATA LOAD
=9083      80 MACROADDL EQU TESTADDRS+2 ;131&132
=9085      81 MACROADDH EQU MACROADDL+2 ;133&134
=9087      82 STARTDATA EQU MACROADDH+2 ;135&136      START OF AUDIO DATA SEGMENT
=9089      83 MATCHQUAL1 EQU STARTDATA+2 ;137&138      QUALITY OF MATCH 1
=908B      84 MATCHQUAL2 EQU MATCHQUAL1+2 ;139&140      QUALITY OF MATCH 2
=908D      85 MATCHQUAL3 EQU MATCHQUAL2+2 ;141&142      QUALITY OF MATCH 3
=908F      86 MATCHQUAL4 EQU MATCHQUAL3+2 ;143&144      QUALITY OF MATCH 4
=9091      87 STOREADDR EQU MATCHQUAL4+2 ;147&148      FILTER OUTPUT
=9093      88 INITIAL EQU STOREADDR+2 ;149      0=INITIAL RECORD
=9094      89 PATTERN EQU INITIAL+1 ;150-151      PATTERN NUMBER (0-3)
=9096      90 CALLSDONE EQU PATTERN+2 ;152-153      NUMBER OF PATTERNS ANALYZED
=9098      91 DAHLFLAG EQU CALLSDONE+2 ;154      FLAG FOR DAHLGREN DATA SENT
=F448      92 MATCHDATA EQU -3000 ;      RAM MATCH DATA

93 ; -----
94 ; RESET AND INTERRUPT VECTORS
95      95 OFFFEH ORG ;RESET VECTOR
96      96 DIGREC FDB
97      97 OFFF2H ORG ;IRQ VECTOR
98      98 RECORD5 FDB
99      99 OFFD6H ORG ;RECEIVE DATA INTERRUPT
100     100 DATAREC FDB
101     101 OFFF6H ORG ;SWI VECTOR
102     102 RECORD5 FDB
103     103 OFFF8H ORG ;ILLEGAL OPCODE TRAP
104     104 DIGREC FDB
105     105 ; -----
106     106 OF800H ORG
107     107 DIGREC EQU $
108     108 SEI EQU
109     109 LDAA #98H ;DISABLE INTERRUPTS
110     110 STAA INIT ;REMAP RAM AND REGISTERS
111     111 LDAA #0 ;RAM STRT=$9000 REGS=$8000
112     112 STAA TMSK2 ;SET TIMER DIVIDER TO E/1
113     113 LDS #90FFH ;MUST BE DONE IN 1ST 64CLKS
114     114 LDAA #10001000B ;LOAD STACK POINTER
115     115 STAA PACTL ;CONFIGURE PORTA BITS 3 & 7
116     116 LDAA #00000100B ;AS OUTPUT TO D/A CONVERTER
117     117 STAA PORTDDDR ;CONFIGURE BITS D0&D1 AS INPUT
;WILL BOTH BE HIGH IMPED FOR DAHLG

```

0018& B6 8039	118	LDA	OPTION		
001B& 8A 80	119	ORAA	#10000000B		;USE EXTERN XTAL CLK & POWER-UP
001D& B7 8039	120	STAA	OPTION		;A/D CONVERTER (BIT 7)
0020& 86 00	121	LDA	#00000000B		;SELECT 8 DATA BITS
0022& B7 802C	122	STAA	SCCR1		
0025& B6 8008	123	LDA	PORTADR		;SET TRANSMIT LINE HIGH=OFF
0028& 84 FB	124	ANDA	#11111011B		;THIS OUTPUT IS INVERTED
002A& B7 8008	125	STAA	PORTADR		
002D& 86 21	126	LDA	#00100001B		;SET BAUD RATE TO 9600
002F& B7 802B	127	STAA	BAUD		;FOR 4.9152 MHz CRYSTAL
0032& 86 FF	128	LDA	#255		;START A/D WARM-UP DELAY
0034& C6 FF	129	LDAB	#255		
0036& 5A	130	DECB			
0037& 26 FD	131	BNE	CNT1		
0039& 4A	132	DECA			
003A& 26 F8	133	BNE	CNT2		;END A/D POWER-UP DELAY
003C& 86 05	134	LDA	#5		
003E& B7 9006	135	STAA	NUMINT		
0041& CC 0000	136	LDD	#0		
0044& B7 8000	137	STAA	PORTADR		
0047& FD 9096	138	STD	CALLSDONE		
004A& FD 9074	139	STD	NUMCALL1		;ZERO OUT ALL CALLS IDENTIFIED
004D& FD 9076	140	STD	NUMCALL3		
0050& FD 900C	141	STD	FILPAR		
0053& B7 9080	142	STAA	PCLOADFLG		
0056& B7 900A	143	STAA	EPOCH		;START WITH EPOCH=0
0059& B7 9078	144	STAA	FREQ		;ZERO OUT FILTER NUMBER
005C& B7 9093	145	STAA	INITIAL		
005F& 86 01	146	LDA	#00000001B		;ENABLE PA0/IC3 INTERRUPT
0061& B7 8022	147	STAA	TMSK1		
0064& B7 8021	148	STAA	TCTL2		;CAPTURE RISING EDGES ONLY
0067& BD 046E&	149	JSR	FULLSEC		
006A& BD 046E&	150	JSR	FULLSEC		
	151				
006D& 8E 90FF	152	NEWSTART	#90FFH		;RELOAD STACK
0070& 07	153	TPA			
0071& 84 7F	154	ANDA	#01111111B		;CLEAR "S" BIT TO ENABLE
0073& 06	155	TAP			;STOP MODE (TESTED 8/27/98=OK)
0074& 0E	156	CLI			;ALLOW INTERRUPTS NOW

```

0075& CF      157      STOP
0076& 01      158      NOP
0077& 20 F4    159      BRA      NEWSTART      ;ALLOW MULTIPLE ANALYSES
160      ;-----
161      ;MOST OF THE FOLLOWING COMMAND INTERPRETERS/DIRECTORS HAVE BEEN
162      ;COMMENTED OUT ON 8/20/98 AS THEY WERE FOR TEST USE ONLY
163      DATAREC      SEI
164      LDAA      #0FFH      ;CLEAR FLAG TO PREVENT
165      STAA      TFLG1      ;CONTINUOUS INTERRUPTS
166      LDAA      SCSR      ;CLEAR RDRF FLAG
167      LDAA      SCDR
168      STAA      INSTRUCT
169      LDAB      PCLOADFLG
170      BEQ      INTERP
171      JSR      INSINE2
172      RTI
173      ;INTERP
174      ;
175      ;
176      ;
177      ;
178      ;NEXTINT1
179      ;
180      ;
181      ;
182      ;NEXTINT2
183      ;
184      ;
185      ;
186      ;NEXTINT3
187      ;
188      ;
189      ;
190      ;NEXTINT4
191      ;
192      ;
193      ;NEXTINT5
194      ;
195      ;

0079& 0F      164      SEI
007A& 86 FF    165      LDAA      #0FFH      ;CLEAR FLAG TO PREVENT
007C& B7 8023  166      STAA      TFLG1      ;CONTINUOUS INTERRUPTS
007F& B6 802E  167      LDAA      SCSR      ;CLEAR RDRF FLAG
0082& B6 802F  168      LDAA      SCDR
0085& B7 900B  169      STAA      INSTRUCT
170      LDAB      PCLOADFLG
171      BEQ      INTERP
172      JSR      INSINE2
173      RTI
174      ;
175      ;
176      ;
177      ;
178      ;NEXTINT1
179      ;
180      ;
181      ;
182      ;NEXTINT2
183      ;
184      ;
185      ;
186      ;NEXTINT3
187      ;
188      ;
189      ;
190      ;NEXTINT4
191      ;
192      ;
193      ;NEXTINT5
194      ;
195      ;

      CMPA      #47H
      BNE      NEXTINT1
      JSR      ECHOID
      RTI
      CMPA      #49H
      BNE      NEXTINT2
      JSR      INSINE
      RTI
      CMPA      #4FH
      BNE      NEXTINT3
      JSR      DUMPAMP
      RTI
      CMPA      #4EH
      BNE      NEXTINT4
      JSR      CHCKNORM
      RTI
      CMPA      #44H
      BNE      NEXTINT5
      JMP      DIGIFILT
      CMPA      #43H
      BNE      NEXTINT6
      JSR      CONFIRM
      ;CHECK SEND ID INSTRU="G"
      ;"I" = INPUT SINE WAVE FROM PC
      ;"O" = DUMP AMPLITUDE
      ;"N"=WATERFALL DUMP
      ;"D"=DIGITAL FILTER
      ;"C"=CONFIRM DUMP

```

```

196 ;NEXTINT6      CMPA    #50H      ; "P"=CHECK MATCH
197 ;              BNE     NEXTINT7
198 ;              JSR     SENDMATCH
199 ;MOVED THESE INTERPRETERS DIRECTLY INTO DAHLGREN ROUTINE
200 ;              CMPA    #53H      ; "S"=SEND DATA
201 ;              JSR     DAHLGREN
202 ;              CMPA    #52H      ; "R"=RESEND DATA
203 ;NEXTINT7      RTI
204 ;
205 ;THE FOLLOWING ROUTINE ALLOWED DOWNLOADS OF SINE WAVE DATA FROM A PC
206 ;TO TEST THE DIGITAL FILTER BANKS AND WAS COMMENTED OUT ON 8/24/98
207 ;INSINE        LDAA    #1
208 ;              STAA    PCLOADFLG
209 ;              LDD     #0
210 ;              STD     TESTADDRS
211 ;              RTS
212 ;INSINE2        LDX     TESTADDRS
213 ;              LDAA    INSTRUCT
214 ;              STAA    0,X
215 ;              INX
216 ;              STX     TESTADDRS
217 ;              CPX     #30720
218 ;              BNE     ENDSINE
219 ;              CLR     PCLOADFLG
220 ;ENDSINE        RTS
221 ;
222 ;THE FOLLOWING SUBROUTINE WAS FOR TEST ONLY AND WAS
223 ;COMMENTED OUT ON 8/24/98
224 ;SENDMATCH      LDD     #MATCHQUAL1 ;SEND FIRST BYTE
225 ;              JSR     ASCII
226 ;              LDD     #MATCHQUAL1 ;SEND SECOND BYTE
227 ;              JSR     ASCII
228 ;              LDD     #0
229 ;              STD     FILPAR
230 ;              RTS
231 ;
232 ;THE FOLLOWING ROUTINE WAS FOR TEST PURPOSES ONLY AND WAS
233 ;COMMENTED OUT ON 8/24/98
234 ;CONFIRM        LDD     TESTADDRS

```



```

235 ; CPD #30720
236 ; BEQ SENDX
237 ; LDD #30750
238 ; SUBD TESTADDRS
239 ; ADDB #30H
240 ; JSR TRANBYTE2
241 ; SENDX LDAB #58H
242 ; JSR TRANBYTE2
243 ; RTS
244 ;
245 ;THE ABS SUBROUTINES ADDS THE FILTERED SAMPLE'S ABSOLUTE VALUES
;IF H IS - SO IS TOTAL
246 ABS LDD MEGAADDH
247 BMI MAKEPLUS
248 BEQ CHECKLOW
249 BRA MACROADD
250 CHECKLOW LDD MEGAADDL
251 BMI MAKEPLUS
252 BRA MACROADD
253 MAKEPLUS LDD #0
254 SUBD MEGAADDH
255 STD MEGAADDH
256 LDD #0
257 SUBD MEGAADDL
258 STD MEGAADDL
259 MACROADD LDD MEGAADDL
260 ADDD MACROADDL
261 BVS MEGAINC2
262 STD MACROADDL
263 BRA ADDHI
264 MEGAINC2 SUBD #-32768
265 STD MACROADDL
266 LDD MACROADDH
267 ADDD #1
268 STD MACROADDH
269 ADDHI LDD MEGAADDH
270 ADDD MACROADDH
271 STD MACROADDH
272 LDD #0
273 STD MEGAADDH
;STORE VALIDATED RESULT

```

```

00D5& FD 907C      274      STD      MEGAADDL
00D8& 39           275      RTS
276 ; -----
277 CHCKNORM      LDD      #AVERAGE+1 ;SEND AVERAGE VALUE FIRST
278              JSR      ASCII
279              LDD      FILPAR
280              CPD      #101
281              BNE      ENDROUT3
282              LDD      #0
283              STD      FILPAR
284 ENDROUT3      RTS
285 ; -----
286 DUMPAMP        LDD      #MATCHDATA
287              JSR      ASCII
288              LDD      FILPAR
289              CPD      #10
290              BNE      ENDROUT2
291              LDD      #0
292              STD      FILPAR
293 ENDROUT2      RTS
294 ; -----
295 ;THE FOLLOWING ROUTINE WAS TO DUMP OUT THE STORED FILTER PARAMETERS
296 ;AND IS FOR TEST PURPOSES ONLY. THIS ROUTINE WAS COMMENTED OUT
297 ;ON 8/24/98
298 ;DUMPPARAM      LDD      #FILT01WGHT
299              JSR      ASCII
300              LDD      FILPAR
301              CPD      #230
302              BNE      ENDROUT1
303              LDD      #0
304              STD      FILPAR
305 ;ENDROUT1      RTS
306 ; -----
307 ASCII          ADDD      FILPAR
308              XGDx      LDD      FILPAR
309              ADDD      #1
310              STD      FILPAR
311              LDAB      0,X
312 NEXTBYTE8

```

;PARAMETERS=10X23=230

0114& C1 00	313	CMPB	#0	
0116& 2D 07	314	BLT	SENDNEG	
0118& 86 00	315	LDA	#0	
011A& FD 9001	316	STD	TEMP1	
011D& 20 22	317	BRA	SENDNUM	
011F& C6 2D	318	LDAB	#45	
0121& BD 0184&	319	JSR	TRANBYTE2	
0124& E6 00	320	LDAB	0,X	
0126& C1 80	321	CMPB	#-128	
0128& 26 0A	322	BNE	OK128	
012A& C6 31	323	LDAB	#31H	
012C& BD 0184&	324	JSR	TRANBYTE2	
012F& CC 001C	325	LDD	#28	
0132& 20 1E	326	BRA	DIV10	
0134& F7 9001	327	STAB	TEMP1	
0137& C6 00	328	LDAB	#0	
0139& F0 9001	329	SUBB	TEMP1	
013C& 86 00	330	LDA	#0	
013E& FD 9001	331	STD	TEMP1	
0141& 1A 83 0064	332	CPD	#100	
0145& 2D 0B	333	BLT	DIV10	
0147& C6 31	334	LDAB	#31H	
0149& BD 0184&	335	JSR	TRANBYTE2	
014C& FC 9001	336	LDD	TEMP1	
014F& 83 0064	337	SUBD	#100	
0152& CE 000A	338	LDX	#10	
0155& 02	339	IDIV		
0156& FD 9001	340	STD	TEMP1	
0159& 8F	341	XGDX		
015A& C3 0030	342	ADDD	#30H	
015D& BD 0184&	343	JSR	TRANBYTE2	
0160& FC 9001	344	LDD	TEMP1	
0163& CB 30	345	ADDB	#30H	
0165& BD 0184&	346	JSR	TRANBYTE2	
0168& C6 20	347	LDAB	#20H	
016A& BD 0184&	348	JSR	TRANBYTE2	
016D& 39	349	RTS		
	350			
	351	ECHOID	#0	

;D=POSITIVE 1-128

;QUOTIENT IN D

;SEND SPACE AFTER EACH
;ASCII VALUE

```

0172& CE 04EB&
0175& E6 00
0177& BD 0184&
017A& 08
017B& 18 08
017D& 18 8C 0011
0181& 26 F2
0183& 39
-----
0184& B6 802E
0187& F7 802F
018A& B6 802E
018D& 84 80
018F& 27 F9
0191& 39
-----
0192& 18 CE 0003
0196& CE 3DEB
0199& 09
019A& 26 FD
019C& 18 09
019E& 26 F6
01A0& 39
-----
01A1& B6 8000
01A4& 88 FF
01A6& B7 8000
01A9& B6 8008
01AC& 88 FF
01AE& B7 8008
01B1& 39
-----
01B2& 86 80
01B4& B7 8000
01B7& 18 CE 000A
01BB& CE AACC
01BE& 09
01BF& 26 FD
01C1& 18 09
-----
352
353 NEXTBYTE7
354
355
356
357
358
359
360 ;
361 TRANBYTE2
362
363 CHECKTRANS
364
365
366
367 ;
368 TRANSDelay
369 NEXT31
370 NEXT30
371
372
373
374
375 ;
376 PORTTOGL
377
378
379
380
381
382
383 ;
384 FULLSEC2
385
386
387 NEXTX11
388 NEXTX10
389
390
-----
#ID
0,X
TRANBYTE2
#17
NEXTBYTE7
-----
LDX
LDAB
JSR
INX
INY
CPY
BNE
RTS
-----
LDA
STAB
LDA
ANDA
BEQ
RTS
-----
LDY
LDX
DEX
BNE
DEY
BNE
RTS
-----
LDA
EORA
STAA
LDA
EORA
STAA
RTS
-----
LDA
STAA
LDY
LDX
DEX
BNE
DEY
-----
SCSR
SCDR
SCSR
#10000000B
CHECKTRANS
-----
#0003H
#03DEBH
NEXT30
NEXT31
-----
PORTAADR
#11111111B
PORTAADR
PORTDADR
#11111111B
PORTDADR
-----
#10000000B
PORTAADR
#000AH
#0AACCH
NEXTX10
-----
;CLEAR SCSR
;BYTE TO BE SENT LOADED
;THIS ROUTINE PROVIDES 250ms
;DELAY FOR TRANSMIT LINE
;LIGHT FAR LED

```

```

01C3& 26 F6
01C5& B6 8000
01C8& 84 7F
01CA& B7 8000
01CD& BD 046E&
01D0& BD 046E&
01D3& 3B

391 BNE NEXTX11
392 LDAA PORTAADR
393 ANDA #01111111B
394 STAA PORTAADR
395 JSR FULLSEC
396 JSR FULLSEC
397 RTI
398 ;-----
399 ;THIS INTERRUPT HANDLER RECORDS 5 SECONDS OF AUDIO AFTER THRESHOLD
400 ;IS REACHED
401 ;THE FOLLOWING THREE LINES LIT A TEST LED DURING RECORDING
402 ;AND WERE COMMENTED OUT ON 8/27/98
403 ; LDAA PORTAADR
404 ; ORAA #00001000B
405 ; STAA PORTAADR
406 ; LDAA PORTDADR
407 ; ANDA #11111011B
408 ; STAA PORTDADR
409 ; LDAA #00000001B
410 ; STAA ADCCTL
411 ; LDX #0
412 ; LDAA #00000001B
413 ; STAA ADCCTL
414 ;
415 ; CHECKAD2
416 ; ANDA #10000000B
417 ; BEQ CHECKAD2
418 ;
419 ; LDAA ADR1
420 ; BGE SUB128
421 ;
422 ; ADDA #127
423 ; ADDA #1
424 ; BRA CONT522
425 ; SUBA #127
426 ; SUBA #1
427 ; BRA CONT522
428 ; STAA 0,X
429 ; INX

;TURN OFF FAR LED
;NO MORE INTERRUPTS FOR
;ABOUT 2 SECONDS

;ZERO OUT RAM A16=PD2
;SET FOR SINGLE CONVERSION
;START CONVERSIONS
;START AT ADDRESS=0
;2 CYC SET FOR SINGLE CONVERSION
;4 CYC START CONVERSIONS
SUBTOTAL = 6 CYCLES
;4CYC CHECK CONVERSION COMPLETE
;2CYC
;3CYC
ABOVE CYC 9 CYCLES DON'T COUNT
;4 CYC
;3 CYC CONVERT 0-255->-128 TO 127
;SUBTOTAL CYCLES=13
;2 CYC
;2 CYC
;3 CYC SUBTOTAL THIS ROUT=20
;2 CYC
;2 CYC
;3 CYC SUBTOTAL THIS ROUT=20
;4 CYC SUBTOT=24
;3 CYC SUBTOT=27

```

```

0204& 8C 7800      CPX      #30720      ; 4 CYC SUBTOT=31
0207& 27 09      BEQ      ENDREC      ; 3 CYC SUBTOT=34  NEED 77CYC
0209& 86 07      LDAA     #7          ; 2 CYC SUBTOT=36
020B& 4A          DECA          ; 2 CYC
020C& 26 FD      BNE      MAINDELAY    ; 3 CYC LOOP=5 5x7=35  CYC ST=71
020E& 20 00      BRA      ADDELAY      ; 2 CYC SUBTOT=74
0210& 20 D2      BRA      NEXTSAMP3    ; 3 CYC SUBTOT=76
0212& B6 8000    LDAA     PORTAADR      ;
0215& 84 F7      ANDA     #11110111B   ;
0217& B7 8000    STAA     PORTAADR      ; TURN OFF LED
021A& BD 021E&   JSR      DIGIFILT     ;
021D& 3B          RTI

021E& CC 0000    LDD      #0          ; ZERO OUT REGISTERS
0221& FD 907C    STD      MEGAADDL      ; ZERO OUT DIGITAL FILTER SUMS
0224& FD 907E    STD      MEGAADDH      ;
0227& FD 9083    STD      MACROADDL     ;
022A& FD 9085    STD      MACROADDH     ; LOAD ENDING ADDRESS OF DATA
022D& B6 900A    LDAA     EPOCH        ;
0230& C6 0C      LDAB     #12          ;
0232& 3D          MUL          ; D=0-108 SO B=0-108
0233& 17          TBA          ; D=(0-108)*256= 0-27,648
0234& 5F          CLR      #3048       ; D=3071+(N*3072)-23
0235& FD 9087    STD      STARTDATA    ; START SAMPLE DATA IN Y REG
0238& C3 0BE8    ADD      ENDDATA      ; 23 STAGES PER FILTER
023B& FD 9008    LDY      #23          ; FREQ= 0-9
023E& 18 FE 9087 LDAB     FREQ         ; D=0 TO 207
0242& C6 17      LDAA     #FILT01WGHT  ; FILTER PARAMETER START IN D
0244& B6 9078    MUL          ; X REG HAS FILT START
0247& 3D          ADD      FILTSTART    ;
0248& C3 04FC&   STD      COUNT        ;
024B& FD 9079    XGDX     0,X          ;
024E& 8F          CLR      0,Y          ;
024F& 7F 9000    LDAA     SIGNMULT      ;
0252& A6 00      LDAB     MEGAADD      ;
0254& 18 E6 00   JSR      COUNT        ;
0257& BD 0416&   JSR      COUNT        ;
025A& BD 03E7&   INC          ;
025D& 7C 9000

```

```

0260& B6 9000      469      LDAA      COUNT
0263& 81 17        470      CMPA      #23
0265& 27 05        471      BEQ       SUMDATA
0267& 08           472      INX
0268& 18 08        473      INY
026A& 20 E6        474      BRA       NEXTELEM
026C& BD 0088&     475      JSR       ABS
026F& 7F 9000      476      CLR       COUNT
0272& 18 BC 9008   477      CPY       ENDDATA
0276& 27 0C        478      BEQ       RESETSAMP
0278& FE 9079      479      LDX      FILTSTART
027B& 18 8F        480      XGDI
027D& 83 0015      481      SUBD      #21
0280& 18 8F        482      XGDI
0282& 20 CE        483      BRA       NEXTELEM
0284& F6 900A      484      LDAB     EPOCH
0287& 86 0A        485      LDAA     #10
0289& 3D           486      MUL
028A& FB 9078      487      ADDB     FREQ
028D& C3 9010      488      ADDD     #FILTAMP
0290& FD 9091      489      STD      STOREADDR
0293& FC 9085      490      LDD      MACROADDH
0296& CE 0014      491      LDX      #20
0299& 02           492      IDIV
029A& 8F           493      XGDX
029B& FE 9091      494      LDX      STOREADDR
029E& E7 00        495      STAB     0,X
029F&             496      ;
029F&             497      ;NEXT SECTION COMMENTED OUT ON 8/19/98 WAS FOR TEST DUMP OF
029F&             498      ;RAW DIGITAL FILTER OUTPUT DATA FOR TEST PURPOSES ONLY
029F&             499      ;NEXTBYTE1      LDD      #MACROADDL ;OUTPUT 4 DATA BYTES
029F&             500      ;                JSR      ASCII
029F&             501      ;                LDD      FILPAR
029F&             502      ;                CPD      #4
029F&             503      ;                BEQ      RESFIL6
029F&             504      ;                BRA      NEXTBYTE1
029F&             505      ;RESFIL6      LDD      #0
029F&             506      ;                STD      FILPAR
029F&             507      ;

```

```

;THIS DATA SAMPLE OF THE
;OUTPUT NOW TO BE SUMMED

```

```

;DONE WITH THIS FILTER
;RESET FILTER ADDRESS
;GET Y INTO D REG
;0-22 SUB 21 INCS BY 1
;NEW ADDRESS IN Y

```

```

;STORE RESULT

```

```

;D=0-90->B=0-90
;B=0-99
;ADD OFFSET

```

```

;DIV BY 20 SHOULD KEEP ALL
;RESULTS<255 MAXIMUM!
;GET QUOTIENT IN D REGISTER

```

```

;NEXT SECTION COMMENTED OUT ON 8/19/98 WAS FOR TEST DUMP OF

```

```

;RAW DIGITAL FILTER OUTPUT DATA FOR TEST PURPOSES ONLY
;NEXTBYTE1      LDD      #MACROADDL ;OUTPUT 4 DATA BYTES

```

```

;ASCII

```

```

;FILPAR

```

```

;#4

```

```

;BEQ      RESFIL6

```

```

;BRA      NEXTBYTE1

```

```

;LDD      #0

```

```

;STD      FILPAR

```

```

02A0& B6 9078          508          LDAA          FREQ
02A3& 4C              509          INCA
02A4& B7 9078          510          STAA          FREQ
02A7& 81 0A           511          CMPA          #10
02A9& 27 03           512          BEQ          RESET815
02AB& 7E 021E&        513          JMP          DIGIFILT
02AE& 7F 9078          514          CLR          FREQ
02B1& 7C 900A          515          INC          EPOCH
02B4& B6 900A          516          LDAA          EPOCH
02B7& 81 0A           517          CMPA          #10
02B9& 27 03           518          BEQ          MAJORRES
02BB& 7E 021E&        519          JMP          DIGIFILT
02BE& 7F 900A          520          CLR          EPOCH
02C1& BD 02FC&        521          JSR          NORMALIZE
02C4& BD 034C&        522          JSR          MATCH
523          ;SKIP MATCH DECISION IN "FINAL" SUBROUTINE AND INSTEAD
524          ;SEND ALL FOUR CORRELATION COEFFICIENTS TO THE DAHLGREN UNIT
525          ;
526          JSR          FINAL
527          JSR          DAHLGREN
528          RTS
529          ;-----
529          ;AS OF AUGUST, 1998, THIS ROUTINE WILL NOT BE USED. ALL FOUR
530          ;CORRELATION VALUES WILL BE SENT OUT TO THE DAHLGREN PROCESSOR
531          FINAL      LDD          MATCHQUAL1
532                      CPD          THRESHOLD
533                      BGT          CHECK2
534                      INC          NUMCALL1
535          CHECK2      LDD          MATCHQUAL2
536                      CPD          THRESHOLD
537                      BGT          CHECK3
538                      INC          NUMCALL2
539          CHECK3      LDD          MATCHQUAL3
540                      CPD          THRESHOLD
541                      BGT          CHECK4
542                      INC          NUMCALL3
543          CHECK4      LDD          MATCHQUAL4
544                      CPD          THRESHOLD
545                      BGT          DONEFINL
546                      INC          NUMCALL4

```

;DONE WITH ALL 10 FREQUENCIES

;RESET FREQUENCY

;REGULAR VALUES=0-9

;NORMALIZE DATA

;CALCULATED CORRELATIONS

;DO FINAL ANALYSIS AND STORE

;AWAIT TRANSFER OPPORTUNITY


```

02FB& 39      547 DONEFINL      RTS
548 ;-----
549 ;SET AVERAGE VALUE TO 38 BY MULTIPLYING ALL VALUES BY C=38/AVERAGE
550 ;NOTE: AVERAGE OF 50 CAUSES PROBLEMS WITH DYNAMIC RANGE OF FILTER OUTPTS
551 NORMALIZE   LDD #0
552 STAA COUNT
553 STD MEGAADDL
554 STD MEGAADDH
555 LDX #FILTAMP
556 LDAA #0
557 LDAB 0,X
558 STD TEMP1
559 JSR MEGAADD
560 INX
561 INC COUNT
562 LDAA COUNT
563 CMPA #100
564 BNE SUMMORE
565 LDD MEGAADDL
566 LDX #100
567 IDIV
568 STX AVERAGE
569 CLR COUNT
570 LDY #FILTAMP
571 LDAA 0,Y
572 LDAB #38
573 MUL
574 LDX AVERAGE
575 IDIV
576 XGDX
577 STAB 0,Y
578 INY
579 INC COUNT
580 LDAA COUNT
581 CMPA #100
582 BNE MORENORM
583 ;-----
584 ;THE FOLLOWING SECTION WAS COMMENTED OUT ON 8/19/98
585 ;THIS SECTION HAD TRANSFERRED THE NORMALIZED FILTER OUTPTS TO THE

```

;VALUE= 0-256 IN D REG
 ;VALUE=0-25,600 (100X256)
 ;AVERAGE=(50*100)/100
 ;X REG=0-256
 ;STORE AVERAGE
 ;D REG NOW HAS VALUE X 256
 ;AIM FOR AVERAGE=38
 ;D REG NOW HAS + VALUE X 38
 ;NOW HAS 0 TO 32,768
 ;QUOTIENT IN X REG=0-256
 ;QUOTIENT IN D
 ;STORE NORMALIZED VALUE

```

586 ;PC AND WAS FOR TEST PURPOSES ONLY
587 ;OUTNORM      JSR   FULLSEC
588 ;             LDD   #FILTAMP
589 ;             JSR   ASCII
590 ;             LDAB  #20H
591 ;             JSR   TRANBYTE2
592 ;             LDD   FILPAR
593 ;             CPD   #100
594 ;             BEQ   ENDNORM
595 ;             BRA   OUTNORM
596 ;ENDNORM      LDD   #0
597 ;             STD   FILPAR
598 ;             RTS
599 ;-----
600 ;THIS ROUTINE WILL DO THE PATTERN MATCH
601 ;STORES ALL 8 BYTES OF MATCHQUAL DATA
602 MATCH      CLR   COUNT
603             LDD   #0
604             STD   PATTERN
605             STD   MEGAADDL
606             STD   MEGAADDH
607             LDY   #MATCHEPOCH1T1
608 SUMMER1     LDX   #FILTAMP
609 SUMPAT      LDAB  0,Y
610             LDAA  #0
611             STD   TEMP1
612             LDAB  0,X
613             LDAA  #0
614             SUBD  TEMP1
615             STD   TEMP1
616             BMI   PLUSIT
617             BRA   ADDIT
618 PLUSIT      LDD   #0
619             SUBD  TEMP1
620             STD   TEMP1
621 ADDIT       LDAA  TEMP2
622             LDAB  TEMP2
623             MUL
624             STD   TEMP1

034B& 39
034C& 7F 9000
034F& CC 0000
0352& FD 9094
0355& FD 907C
0358& FD 907E
035B& 18 CE 05E1&
035F& CE 9010
0362& 18 E6 00
0365& 86 00
0367& FD 9001
036A& E6 00
036C& 86 00
036E& B3 9001
0371& FD 9001
0374& 2B 02
0376& 20 09
0378& CC 0000
037B& B3 9001
037E& FD 9001
0381& B6 9002
0384& F6 9002
0387& 3D
0388& FD 9001

;ONE SEC DELAY BETWEEN BYTES
;ADDRESS=FILPAR+#FILTAMP
;SENT BY ASCII ROUTINE
;SEND A SPACE
;CHECK FOR 100 PARAMETERS
;SENT
;START OF TEMPLATE DATA
;START OF NORMALIZED FILTER
;TEMPLATE DATA INTO LOW BYTE
;ZERO OUT HIGH BYTE
;OUTPUT DATA INTO LOW BYTE
;ZERO OUT HIGH BYTE
;RESULT IN D REG
;STORE RESULT NOW
;NEGATE RESULT
;SO THAT ITS ALWAYS POS.
;LOW BYTE (0-255)
;LOW BYTE (0-255)
;STORE SQUARED RESULT

```



```
03E7& FC 9001      664 MEGAADD      LDD      TEMP1
03EA& F3 907C      665      MEGAADDL
03ED& 29 04       666      MEGAINC
03EF& FD 907C      667      MEGAADDL
03F2& 39          668      RTS
03F3& 2B 11       669 MEGAINC      BMI      FLIPIT1
03F5& 83 8000     670      SUBD      #-32768
03F8& FD 907C     671      STD      MEGAADDL
03FB& FC 907E     672      LDD      MEGAADDH
03FE& 83 0001     673      SUBD      #1
0401& FD 907E     674      STD      MEGAADDH
0404& 20 0F       675      BRA      ENDMEG
0406& 83 8000     676 FLIPIT1      SUBD      #-32768
0409& FD 907C     677      STD      MEGAADDL
040C& FC 907E     678      LDD      MEGAADDH
040F& C3 0001     679      ADD      #1
0412& FD 907E     680      STD      MEGAADDH
0415& 39          681 ENDMEG      RTS
682 :-----
0416& B7 9001     683 SIGNMULT      STAA     TEMP1
0419& F7 9002     684      STAB     TEMP2
041C& F8 9001     685      EORB     TEMP1
041F& C4 80       686      ANDB     #10000000B
0421& F7 907B     687      STAB     SIGN
0424& F6 9001     688      LDAB     TEMP1
0427& 2B 07       689      BMI      NEGATE
0429& F6 9002     690 TESTIT2      LDAB     TEMP2
042C& 2B 15       691      BMI      NEGATE1
042E& 20 24       692      BRA      MULT1
0430& C1 80       693 NEGATE      CMPB     #-128
0432& 26 07       694      BNE      GOON81
0434& 86 7F       695      LDAA     #127
0436& B7 9001     696      STAA     TEMP1
0439& 20 EE       697      BRA      TESTIT2
043B& 86 00       698 GOON81      LDAA     #0
043D& 10          699      SBA
043E& B7 9001     700      STAA     TEMP1
0441& 20 E6       701      BRA      TESTIT2
0443& C1 80       702 NEGATE1      CMPB     #-128

;RESULT WAS+/LOOKS -
;STORE VALIDATED RESULT

;RESULT ACCUM A
```

```

0445& 26 07      703      BNE      GOON81A
0447& 86 7F      704      LDAA     #127
0449& B7 9002    705      STAA     TEMP2
044C& 20 06      706      BRA      MULT1
044E& 86 00      707      LDAA     #0
0450& 10        708      SBA
0451& B7 9002    709      STAA     TEMP2
0454& B6 9001    710      LDAA     TEMP1
0457& F6 9002    711      LDAB     TEMP2
045A& 3D        712      MUL
045B& FD 9001    713      STD      TEMP1
045E& B6 907B    714      LDAA     SIGN
0461& 26 01      715      BNE      NEGATEF
0463& 39        716      RTS
0464& CC 0000    717      LDD      #0
0467& B3 9001    718      SUBD     TEMP1
046A& FD 9001    719      STD      TEMP1
046D& 39        720      RTS
046E& 18 CE 000A 721      ;-----
0472& CE AACC    722      FULLSEC #000AH
0475& 09        723      NEXT11 #0AACCH
0476& 26 FD      724      NEXT10
0478& 18 09      725      BNE      NEXT10
047A& 26 F6      726      DEY
047C& 39        727      BNE      NEXT11
047D& 39        728      RTS
047E& 39        729      ;-----
730 ;THIS ROUTINE SENDS THE DATA TO THE DAHLGREN PROCESSOR UPON REQUEST
731 ;1) IT IS EXECUTED AS AN INFINITE LOOP AFTER AN AUDIO ANALYSIS
732 ;2) IT FIRST CHECKS PA0 (HANDSHAKE IN) TO SEE IF IT IS HIGH
733 ;3) IF IT IS HIGH, THE DACIS ENABLES THE SERIAL OUPUT LINE
734 ;4) THE DACIS THEN WAITS AND LOOKS FOR AN "S" (SEND DATA)
735 ;5) THE DACIS THEN SENDS THE FOUR SERIAL SINGLE BYTE VALUES CORRESPONDING
736 ; TO THE FOUR TEMPLATE CORRELATIONS FOR THIS WOLF CALL
737 ;6) THE DACIS THEN LOOPS FOR APPROXIMATELY ONE HALF SECOND CHECKING TO
738 ; SEE IF THE DAHLGREN PROCESSOR HAS SENT AN "R" FOR RESEND THE DATA
739 ;7) AFTER THE ONE SECOND PASSES, THE ROUTINE RESETS COUNTERS AND
740 ; EXECUTES AN RTI
741 ;8) THIS ROUTINE MODIFIED TO SEND ALL 8 BYTES OF MATCH DATA

```

047D& 7F 9098	742 DAHLGREN	CLR	DAHLFLAG	;CLEAR DAHLGREN FLAG
0480& B6 8000	743	LDAA	PORTAADR	;UNTIL DAT SENT ONCE
0483& 8A 08	744	ORAA	#00001000B	;SET PA3 HIGH
0485& B7 8000	745	STAA	PORTAADR	
0488& B6 8000	746 DAHLGREN2	LDAA	PORTAADR	
048B& 84 01	747	ANDA	#00000001B	;CHECK BIT PA0
048D& 26 02	748	BNE	SENDDAHL1	
048F& 20 F7	749	BRA	DAHLGREN2	;CHECK HNDSHK IN AGAIN
0491& 86 0C	750 SENDDAHL1	LDAA	#00001100B	;ENABLE TRANS/REC
0493& B7 802D	751	STAA	SCCR2	;NO TRANS. OR REC. INT.
0496& CE 6C81	752 SENDDAHL2	LDX	#27777	;27,777 X 18uS =0.5 SEC
0499& B6 802E	753 SENDDAHL3	LDAA	SCSR	;4 CYC
049C& 84 20	754	ANDA	#00100000B	;2 CYC CHECK RECEIVE DATA
049E& 26 0C	755	BNE	CHECKCOMM1	;3 CYC REGISTER FULL
04A0& 09	756	DEX	CHECKFLAG1	;3 CYC
04A1& 27 02	757	BEQ	SENDDAHL3	;3 CYC TRY AGAIN
04A3& 20 F4	758	BRA		
	759 ; TOTAL= 18 CYCLES			
04A5& B6 9098	760 CHECKFLAG1	LDAA	DAHLFLAG	;CHECK FOR DATA SENT ONCE
04A8& 26 2E	761	BNE	ENDDAHLGREN	;QUIT IF SENT ONCE
04AA& 20 EA	762	BRA	SENDDAHL2	;MUST TRANSMIT ONCE!
04AC& B6 802F	763 CHECKCOMM1	LDAA	SCDR	;GET DATA INTO A REG
04AF& F6 802E	764	LDAB	SCSR	;CLEAR RDRF FLAG
04B2& 81 53	765	CMPA	#53H	; "S"=SEND DATA
04B4& 27 06	766	BEQ	SEND4BYTES	
04B6& 81 52	767	CMPA	#52H	; "R"=RESEND DATA
04B8& 27 02	768	BEQ	SEND4BYTES	
04BA& 20 DA	769	BRA	SENDDAHL2	;CONTINUE LOOKING
04BC& 86 01	770 SEND4BYTES	LDAA	#1	
04BE& B7 9098	771	STAA	DAHLFLAG	
04C1& 18 CE 0004	772	LDY	#4	;FIRST MATCH BYTE
04C5& CE 9089	773	LDX	#MATCHQUAL1	
04C8& 08	774	INX		
04C9& E6 00	775 SMDMORE1	LDAB	0,X	;LOAD DATA INTO B REG
04CB& BD 0184&	776	JSR	TRANBYTE2	;SEND DATA
04CE& 08	777	INX		
04CF& 08	778	INX		
04D0& 18 09	779	DEY		
04D2& 27 C2	780	BEQ	SENDDAHL2	;DO IT AGAIN?

```

04D4& 20 F3          781          BRA          SNDMORE1
04D6& 20 00          782          BRA          ENDAHLGREN
04D8& 86 00          783          LDAA          #00000000B ;DISABLE TRANS/REC
04DA& B7 802D        784          STAA          SCCR2      ;NO TRANS. OR REC. INT.
04DD& 7F 9098        785          CLR          DAHLFLAG  ;NOT NEEDED HERE BUT CLR
04E0& B6 8000        786          LDAA          PORTAADR ;ANYWAY
04E3& 84 F7          787          ANDA          #11110111B ;SET PA3 BACK LOW
04E5& B7 8000        788          STAA          PORTAADR
04E8& 7E 006D&      789          JMP          NEWSTART ;RESTART FOR NEXT CALL
04EB& 42 49 52 44 42 790          ;-----
04F0& 4F 52 4E 45 20 791 ID          FCB 42H,49H,52H,44H,42H,4FH,52H,4EH,45H,20H ;BIRDBORNE
04F5& 55 4E 49 54 20 792          FCB 55H,4EH,49H,54H,20H,20H,31H ;UNIT__1
04FA& 20 31

04FC& AB AE B9 CB E3          793 ; DIGITAL FILTER WEIGHTS STORED HERE
0501& 00 1E 3C 57 6C          794 ; FOR THE FOLLOWING CENTER FREQUENCIES:
0506& 7A              795 ; 250Hz,500Hz,750Hz,1000Hz,1250Hz,1500Hz,1750Hz,2000Hz,2250Hz,2500Hz
0507& 7F 7A 6C 57 3C          796 FILT01WGHT FCB -85,-82,-71,-53,-29,0,30,60,87,108,122
050C& 1E 00 E3 CB B9          797          FCB 127,122,108,87,60,30,0,-29,-53,-71,-82,-85
0511& AE AB          798 FILT02WGHT FCB 75,46,-1,-53,-96,-115,-103,-61,0,62,109,127
0513& 4B 2E FF CB A0          799          FCB 109,62,0,-61,-103,-115,-96,-53,-1,46,75
0518& 8D 99 C3 00 3E          800 FILT03WGHT FCB -62,0,70,105,77,-1,-84,-122,-88,0,89,127
051D& 6D 7F          801          FCB 89,0,-88,-122,-84,-1,77,105,70,0,-62
051F& 6D 3E 00 C3 99          802 FILT04WGHT FCB 43,-47,-100,-53,54,114,59,-61,-124,-63,63,127
0524& 8D A0 CB FF 2E          803          FCB 63,-63,-124,-61,59,114,54,-53,-100,-47,43
0529& 4B
052A& C2 00 46 69 4D
052F& FF AC 86 A8 00
0534& 59 7F
0536& 59 00 A8 86 AC
053B& FF 4D 69 46 00
0540& C2
0541& 2B D1 9C CB 36
0546& 72 3B C3 84 C1
054B& 3F 7F
054D& 3F C1 84 C3 3B

```

0557& 2B					
0558& E9	51 46 CB 95				
055D& 00	72 3C A8 93				
0562& 20	7F				
0564& 20	93 A8 3C 72				
0569& 00	95 CB 46 51				
056E& E9					
056F& FF	A2 00 69 FF	804	FILT05WGHT	FCB	-23, 81, 70, -53, -107, 0, 114, 60, -88, -109, 32, 127
0574& 8D	00 79 FF 82				
0579& 00	7F				
057B& 00	82 FF 79 00				
0580& 8D	FF 69 A2 FF				
0585& 16	51 B9 CB 6A	805		FCB	32, -109, -88, 60, 114, 0, -107, -53, 70, 81, -23
058A& FF	8D 3C 57 93				
058F& DF	7F				
0591& DF	93 57 3C 8D	806	FILT06WGHT	FCB	-1, -94, 0, 105, -1, -115, 0, 121, -1, -126, 0, 127
0596& FF	6A CB B9 51				
059B& 16					
059C& D4	D1 63 CB C9				
05A1& 72	C4 C3 7B C1	807		FCB	0, -126, -1, 121, 0, -115, -1, 105, -94, -1
05A6& C0	7F				
05A8& C0	C1 7B C3 C4	808	FILT07WGHT	FCB	22, 81, -71, -53, 106, -1, -115, 60, 87, -109, -33, 127
05AD& 72	C9 CB 63 D1				
05B2& D4					
05B3& 3D	FF B9 69 B2	809		FCB	-33, -109, 87, 60, -115, -1, 106, -53, -71, 81, 22
05B8& 00	53 86 57 FF				
05BD& A6	7F				
05BF& A6	FF 57 86 53	810	FILT08WGHT	FCB	-44, -47, 99, -53, -55, 114, -60, -61, 123, -63, -64, 127
05C4& 00	B2 69 B9 FF				
05C9& 3D		811		FCB	-64, -63, 123, -61, -60, 114, -55, -53, 99, -47, -44
05CA& B4	2E FF CB 5F				
05CF& 8D	66 C3 00 3E	812	FILT09WGHT	FCB	61, -1, -71, 105, -78, 0, 83, -122, 87, -1, -90, 127
05D4& 92	7F				
05D6& 92	3E 00 C3 66	813		FCB	-90, -1, 87, -122, 83, 0, -78, 105, -71, -1, 61
05DB& 8D	5F CB FF 2E				
05E0& B4		814	FILT10WGHT	FCB	-76, 46, -1, -53, 95, -115, 102, -61, 0, 62, -110, 127
		815		FCB	-110, 62, 0, -61, 102, -115, 95, -53, -1, 46, -76
		816 ;			
		817 ;			

PARAMETERS FOR CORRELATION MATCHING ALGORITHM STORED HERE


```
05E1& 7E 74 55 1E 15
05E6& 12 11 07 0C 09
05EB& 84 79 53 28 14
05F0& 12 09 08 0C 0A
05F5& 71 80 5C 22 16
05FA& 12 0B 0B 0D 09
05FF& 77 77 4B 21 14
0604& 11 0A 08 0D 09
0609& 7B 6D 48 1E 12
060E& 11 0A 06 0C 09
0613& 82 6D 42 1A 14
0618& 12 0E 07 0D 09
061D& 64 60 45 1A 13
0622& 12 08 06 0C 09
0627& 55 56 3F 19 12
062C& 10 06 06 0C 07
0631& 4A 3F 2B 18 11
0636& 0F 06 05 0C 05
063B& 53 3E 31 1B 11
0640& 0F 05 04 0C 06

0645& 6C 5C 36 1C 1B
064A& 19 1D 17 12 09
064F& 74 5E 44 22 1E
0654& 1D 24 19 14 0C
0659& 7F 5F 39 1F 1D
065E& 17 14 0E 11 0A
0663& 68 53 31 1C 1B
0668& 18 1B 11 11 0A
066D& 62 54 39 1D 19
0672& 18 17 14 13 0A
0677& 66 48 2F 1C 1A
067C& 15 0F 0C 10 08
0681& 5A 33 26 1A 16
0686& 13 0F 0A 10 08
068B& 53 5B 37 20 1A

819 ;WOLF HOWL COMPOSITE TEMPLATE
820 MATCHEPOCH1T1 FCB 126,116,85,30,21,18,17,7,12,9
821 MATCHEPOCH2T1 FCB 132,121,83,40,20,18,9,8,12,10
822 MATCHEPOCH3T1 FCB 113,128,92,34,22,18,11,11,13,9
823 MATCHEPOCH4T1 FCB 119,119,75,33,20,17,10,8,13,9
824 MATCHEPOCH5T1 FCB 123,109,72,30,18,17,10,6,12,9
825 MATCHEPOCH6T1 FCB 130,109,66,26,20,18,14,7,13,9
826 MATCHEPOCH7T1 FCB 100,96,69,26,19,18,8,6,12,9
827 MATCHEPOCH8T1 FCB 85,86,63,25,18,16,6,6,12,7
828 MATCHEPOCH9T1 FCB 74,63,43,24,17,15,6,5,12,5
829 MATCHEPOCH10T1 FCB 83,62,49,27,17,15,5,4,12,6

830 ;FOR ANIMAL CALL TYPE2
831 ;WOLF GROWL COMPOSITE TEMPLATE
832 MATCHEPOCH1T2 FCB 108,92,54,28,27,25,29,23,18,9
833 MATCHEPOCH2T2 FCB 116,94,68,34,30,29,36,25,20,12
834 MATCHEPOCH3T2 FCB 127,95,57,31,29,23,20,14,17,10
835 MATCHEPOCH4T2 FCB 104,83,49,28,27,24,27,17,17,10
836 MATCHEPOCH5T2 FCB 98,84,57,29,25,24,23,20,19,10
837 MATCHEPOCH6T2 FCB 102,72,47,28,26,21,15,12,16,8
838 MATCHEPOCH7T2 FCB 90,51,38,26,22,19,15,10,16,8
839 MATCHEPOCH8T2 FCB 83,91,55,32,26,21,19,13,18,13
```

```

0695& 7E 50 2F 1D 1C
069A& 14 13 0E 10 09
069F& 65 4A 34 1D 19
06A4& 17 15 12 11 09

06A9& 3E 3B 3D 30 3E
06AE& 28 31 25 14 0D
06B3& 3F 3F 30 28 2E
06B8& 1D 2A 1E 11 0A
06BD& 3A 3B 33 2F 33
06C2& 1E 28 1A 10 0B
06C7& 41 4E 33 36 30
06CC& 1E 2B 1C 12 0C
06D1& 35 42 2C 38 2B
06D6& 18 1B 13 0D 08
06DB& 30 36 32 31 32
06E0& 1B 1F 17 0F 0A
06E5& 30 39 24 24 23
06EA& 17 1B 14 0E 08
06EF& 32 3E 34 2E 30
06F4& 1F 25 20 12 0B
06F9& 33 49 2C 26 25
06FE& 18 1C 16 0E 08
0703& 43 5E 33 2C 2A
0708& 1C 21 1A 0D 0A

070D& 46 66 24 29 1A
0712& 27 26 0E 10 0A
0717& 4A 6B 22 2B 1E
071C& 29 3F 81 16 0D
0721& 40 1E 14 15 10
0726& 0F 08 07 0D 06
072B& 41 37 1B 18 13
0730& 15 11 09 0F 07
0735& 5B 5E 2D 20 1C
073A& 28 30 17 11 0B

840 MATCHEPOCH9T2 FCB 126,80,47,29,28,20,19,14,16,9
841 MATCHEPOCH10T2 FCB 101,74,52,29,25,23,21,18,17,9
842 ; FOR ANIMAL CALL TYPE3
843 ; WOLF BARK COMPOSITE TEMPLATE
844 MATCHEPOCH1T3 FCB 62,59,61,48,62,40,49,37,20,13
845 MATCHEPOCH2T3 FCB 63,63,48,40,46,29,42,30,17,10
846 MATCHEPOCH3T3 FCB 58,59,51,47,51,30,40,26,16,11
847 MATCHEPOCH4T3 FCB 65,78,51,54,48,30,43,28,18,12
848 MATCHEPOCH5T3 FCB 53,66,44,56,43,24,27,19,13,8
849 MATCHEPOCH6T3 FCB 48,54,50,49,50,27,31,23,15,10
850 MATCHEPOCH7T3 FCB 48,57,36,36,35,23,27,20,14,8
851 MATCHEPOCH8T3 FCB 50,62,52,46,48,31,37,32,18,11
852 MATCHEPOCH9T3 FCB 51,73,44,38,37,24,28,22,14,8
853 MATCHEPOCH10T3 FCB 67,94,51,44,42,28,33,26,13,10
854 ; FOR ANIMAL CALL TYPE 4
855 ; WOLF WHINE COMPOSITE TEMPLATE
856 MATCHEPOCH1T4 FCB 70,102,36,41,26,39,38,14,16,10
857 MATCHEPOCH2T4 FCB 74,107,34,43,30,41,63,129,22,13
858 MATCHEPOCH3T4 FCB 64,30,20,21,16,15,8,7,13,6
859 MATCHEPOCH4T4 FCB 65,55,27,24,19,21,17,9,15,7
860 MATCHEPOCH5T4 FCB 91,94,45,32,28,40,48,23,17,11

```

073F& AB A6 4E 2F 37
0744& 36 36 38 1B 16

861 MATCHEPOCH6T4 FCB 171,166,78,47,55,54,54,56,27,22

Avocet 6811 Assembler v2.23, #01301 Chip=68HC11

8/27/98 15:19:01
Page 25

0749& 46 2A 1A 22 1D
074E& 19 1A 11 13 0B
0753& 41 21 18 16 11
0758& 0F 0A 08 0D 08
075D& 34 1B 16 17 12
0762& 24 10 11 1E 15
0767& 4C 26 15 16 14
076C& 13 0F 0F 13 17
0771& 0B B8

862 MATCHEPOCH7T4 FCB 70,42,26,34,29,25,26,17,19,11

863 MATCHEPOCH8T4 FCB 65,33,24,22,17,15,10,8,13,8

864 MATCHEPOCH9T4 FCB 52,27,22,23,18,36,16,17,30,21

865 MATCHEPOCH10T4 FCB 76,38,21,22,20,19,15,15,19,23

866 THRESHOLD FCB 0BH,0B8H ;THRESHOLD=3000

867 ;TOP OF EEPROM IS 07BFH=1983 BYTES

868 ;

869
END

Enclosure 3
BASIC Software for PC

```

REM
REM      BIRDBORN PC TEST PROGRAM
REM      WOLFTEST.BAS
REM      TESTS DAHLGREN INTERFACE
REM
REM      WRITTEN BY: PROTAGORAS CUTCHIS
REM
REM      VERSION 1.2 - WEDNESDAY-AUGUST 26, 1998
REM      TIME 3:40 PM
REM
REDU: DIM DIGITS%(20)
      ERRORS = 0
      OPEN "COM1:9600,N,8,1" FOR RANDOM AS #1 LEN = 256
      REM CONFIGURE RS232 PORT
      GOSUB INITDATE
      REM DEFINE FUNCTION KEYS ...
      KEY ON
      KEY 10, "QUIT"
      KEY(10) ON
      ON KEY(10) GOSUB SEEYA
      HELLO: REM PRINT HELLO SCREEN ...
      CLS
      PRINT "PRESS 'F10' KEY AT ANY TIME TO QUIT THE PROGRAM "
      PRINT "      IF THE PROGRAM IS WAITING FOR YOU TO ENTER A NUMBER";
      PRINT " AND YOU WANT TO QUIT"
      PRINT " PRESS ENTER AFTER PRESSING THE 'F10' KEY"
      LOCATE 7, 1
      GOTO TAGSTUF
STALL: REM CAUSES PROGRAMS TO WAIT 'TIMEDELAY' SECONDS TO ALLOW RECEIVER
      REM OR THE RS232 INTERFACE TO COMPLETE AN ACTION
      FOR XX = 1 TO 100
      FOR YY = 1 TO 20
      NEXT YY
      NEXT XX
      ' ADDS SMALL DELAY BEFORE CONTINUING TO ENSURE CURRENTTIMER
      ' IS GREATER THAN NOW
      RETURN
WEIGHT14: CURRENTTIMER = TIMER
      IF CURRENTTIMER > 86399.5 THEN GOTO WEIGHT14
      IF CURRENTTIMER < NOW THEN
      REM THIS IS TRUE ONLY WHEN TIMER RESETS TO 0 AT MIDNIGHT
      NOW = 0
      END IF
      IF CURRENTTIMER - NOW < TIMEDELAY THEN GOTO WEIGHT14
      RETURN

```

```

REM #####
INITDATE: REM CONVERTS DATE AND TIME TO FILE NAME
' FILE NAME = DDD_HHMM.DAT WHERE
' DDD = DAY OF THE YEAR (1-365). AND HHMM = HOURS-MINUTES
DAZE$ = DATE$
DAY = VAL(MID$(DAZE$, 4, 2))
MONTH = VAL(MID$(DAZE$, 1, 2))
YEARS = VAL(MID$(DAZE$, 7, 4))
IF MONTH = 1 THEN DAYNUM = 0
IF MONTH = 2 THEN DAYNUM = 31
IF MONTH = 3 THEN DAYNUM = 59
IF MONTH = 4 THEN DAYNUM = 90
IF MONTH = 5 THEN DAYNUM = 120
IF MONTH = 6 THEN DAYNUM = 151
IF MONTH = 7 THEN DAYNUM = 181
IF MONTH = 8 THEN DAYNUM = 212
IF MONTH = 9 THEN DAYNUM = 243
IF MONTH = 10 THEN DAYNUM = 273
IF MONTH = 11 THEN DAYNUM = 304
IF MONTH = 12 THEN DAYNUM = 334
IF YEARS MOD 4 = 0 AND DAYNUM >= 59 THEN DAYNUM = DAYNUM + 1
DAYNUM = DAYNUM + DAY
DAYNUMS$ = MID$(STR$(DAYNUM), 2, 3)
HOURS$ = MID$(TIME$, 1, 2)
MINS$ = MID$(TIME$, 4, 2)
FILENAME$ = DAYNUMS$ + "_" + HOURS$ + MINS$

RETURN
REM #####
TAGSTUF: PRINT "IT IS RECOMMENDED THAT THE SYSTEM ID BE CHECKED"
PRINT "TO CONFIRM SYSTEM COMMUNICATIONS"
PRINT "DO YOU WISH TO CONFIRM SYSTEM ID?"
PRINT "ENTER Y/N"

TRYAGAIN1: TAGVAR$ = INKEY$
IF TAGVAR$ = "Y" OR TAGVAR$ = "y" THEN GOTO GETID
IF TAGVAR$ = "N" OR TAGVAR$ = "n" THEN CLS : GOTO NEXTTEST1
GOTO TRYAGAIN1

GETID:
GOSUB GETID2
GOTO NEXTTEST1

GETID2:
INNY$ = INPUT$(LOC(1), #1)
EFLINE$ = "G"
PRINT #1, EFLINE$
GOSUB STALL
RS232 = 99
INNY$ = INPUT$(LOC(1), #1)
RS232 = 11

```

```

PRINT INNY$
RETURN
NEXTTEST1: NORMFLAG = 0
PRINT "TEST OPTIONS ARE:"
PRINT " 1) DUMP ALL FILTER WEIGHTS"
PRINT " 2) DO A SINGLE TIME EPOCH FREQUENCY TEST"
PRINT " 3) DO A MULTI-EPOCH FREQUENCY TEST"
PRINT " 4) RECONFIRM UNIT ID CODE"
PRINT " 5) TEST SIGNED MULTIPLY ROUTINE"
PRINT " 6) TEST DIGITAL MEGA ADDITION ROUTINE"
PRINT " 7) TEST A MULTI-TIME EPOCH MULTI-FREQUENCY FILTER RESPONSE"
PRINT " 8) TEST THE NORMALIZATION ROUTINE"
PRINT " 9) TEST DIGITAL FILTERS AND NORMALIZATION WITH ACTUAL RECORDED DATA"
PRINT " A) ANALYZE A RECORDING AND DUMP NORMALIZED PARAMETERS"
PRINT " B) DO A MATCH AGAINST A RECORDED REFERENCE"
PRINT " C) MATCH 2 STORED DATA FILES"
PRINT " D) TEST INTERFACE WITH DAHLGREN GPS/ARGOS UNITS"
PRINT " E) TEST DAHLGREN INTERFACE AND TRANSFER 8 BYTES"
PRINT "ENTER 1-9 OR A-E"

TRYAGAIN2: TAGVAR$ = INKEY$
IF TAGVAR$ = "1" THEN CLS : GOTO DUMPWGHT
IF TAGVAR$ = "2" THEN CLS : GOTO FILTEST
IF TAGVAR$ = "3" THEN CLS : GOTO WATERFALL
IF TAGVAR$ = "4" THEN CLS : GOSUB GETID2: GOTO NEXTTEST1
IF TAGVAR$ = "5" THEN CLS : GOTO MULTIPLY
IF TAGVAR$ = "6" THEN CLS : GOTO MEGATEST
IF TAGVAR$ = "7" THEN CLS : GOTO MULTTEST
IF TAGVAR$ = "8" THEN CLS : GOTO NORMAL
IF TAGVAR$ = "9" THEN CLS : GOTO AUDIO
IF TAGVAR$ = "A" OR TAGVAR$ = "a" THEN CLS : GOTO ANALYZE1
IF TAGVAR$ = "B" OR TAGVAR$ = "b" THEN CLS : GOTO CAPTURE
IF TAGVAR$ = "C" OR TAGVAR$ = "c" THEN GOTO MATCH1
IF TAGVAR$ = "D" OR TAGVAR$ = "d" THEN CLS : DAHLFLAG = 0: GOTO DAHLGREN
IF TAGVAR$ = "E" OR TAGVAR$ = "e" THEN CLS : DAHLFLAG = 1: GOTO DAHLGREN
GOTO TRYAGAIN2

MATCH1:
PRINT "ENTER FIRST FILE NAME."
INPUT FILE1$
PRINT "ENTER SECOND FILE NAME."
INPUT FILE2$
OPEN FILE1$ FOR INPUT AS #3
FOR L = 1 TO 100
INPUT #3, A(L)
NEXT L
CLOSE #3
OPEN FILE2$ FOR INPUT AS #3
FOR L = 1 TO 100

```

```

INPUT #3, B(L)
NEXT L
CLOSE #3
DIFF1 = 0: DIFF2 = 0
FOR L = 1 TO 100
  DIFF1 = DIFF1 + ABS(A(L) - B(L))
NEXT L
FOR L = 1 TO 9
  FOR K = 0 TO 9
    INDEX = L + (K * 10)
    DIFF2 = DIFF2 + ABS(A(INDEX) - B(INDEX))
  NEXT K
NEXT L
PRINT "MATCH DIFFERENCE BASED ON STRAIGHT"
PRINT "100 PARAMETER TIME/FREQ DISTRIBUTION IS: "; DIFF1
PRINT " "
PRINT "MATCH DIFFERENCE BASED ON 20 PARAMETER FREQ/VARIANCE"
PRINT "DISTRIBUTION IS: "; DIFF2
TAGVAR$ = INKEY$
IF TAGVAR$ = " " THEN GOTO NEXTTEST1
GOTO TA911A
ANALYZE1: NORMFLAG = 1
PRINT "START CAPTURE"
PRINT "START TIME IS: "; TIME$: TSTRT = 3600 * VAL(LEFT$(TIME$, 2)) + 60 * VAL(MID$(TIME$, 4, 2)) +
VAL(RIGHT$(TIME$, 2)) >= TSTRT + 800 THEN GOTO
NORMAL
CHCKTIME1: IF 3600 * VAL(LEFT$(TIME$, 2)) + 60 * VAL(MID$(TIME$, 4, 2)) + VAL(RIGHT$(TIME$, 2)) >= TSTRT + 800 THEN GOTO
NORMAL
GOTO CHCKTIME1
DAHLGREN: PRINT "HIT Y KEY AND ENTER WHEN YELLOW LIGHT STAYS LIT"
PRINT "THIS WILL BE APPROXIMATELY 15 MINUTES AFTER CAPTURE"
TAGVAR$ = INKEY$
IF TAGVAR$ = "Y" OR TAGVAR$ = "y" THEN GOTO DAHL2
GOTO GETY1
PRINT " "
PRINT " "
PRINT "HIT MOMENTARY PUSHBUTTON SWITCH ON PROTOTYPE"
PRINT "AND HOLD DOWN WHILE HITTING X AND ENTER."
TAGVAR$ = INKEY$
IF TAGVAR$ = "X" OR TAGVAR$ = "x" THEN GOTO DAHL3
FOR L = 1 TO 200
  NEXT L
GOTO GETX1
PRINT "OK"
INNY$ = INPUT$(LOC(1), #1)

```



```

EFLINE$ = "S"
PRINT #1, EFLINE$
RS232 = 99
FOR L = 1 TO 200
NEXT L
INNY$ = INPUT$(LOC(1), #1)
RS232 = 11
EFLINE$ = "R"
PRINT #1, EFLINE$
RS232 = 99
FOR L = 1 TO 500
NEXT L
INNY2$ = " "
INNY2$ = INPUT$(LOC(1), #1)
PRINT "ON SEND REQUEST: DATA=: ";
CHAR$ = LEFT$(INNY$, 1)
GOSUB ASCIIHEX
PRINT A$; " ";
CHAR$ = MID$(INNY$, 2, 1)
GOSUB ASCIIHEX
PRINT A$; " ";
CHAR$ = MID$(INNY$, 3, 1)
GOSUB ASCIIHEX
PRINT A$; " ";
IF DAHLFLAG = 0 THEN GOTO RIGHT1
CHAR$ = MID$(INNY$, 4, 1)
GOSUB ASCIIHEX
PRINT A$; " ";
CHAR$ = MID$(INNY$, 5, 1)
GOSUB ASCIIHEX
PRINT A$; " ";
CHAR$ = MID$(INNY$, 6, 1)
GOSUB ASCIIHEX
PRINT A$; " ";
CHAR$ = MID$(INNY$, 7, 1)
GOSUB ASCIIHEX
PRINT A$; " ";
CHAR$ = RIGHT$(INNY$, 1)
GOSUB ASCIIHEX
PRINT A$; " ";
PRINT " BYTES="; LEN(INNY$)
RIGHT1:
PRINT "ON RESEND REQUEST: DATA=: ";
RETRAN1:
CHAR$ = LEFT$(INNY2$, 1)
GOSUB ASCIIHEX
PRINT A$; " ";
CHAR$ = MID$(INNY2$, 2, 1)

```

```

GOSUB ASCIHEX
PRINT A$; " ";
CHAR$ = MID$(INNY2$, 3, 1)
GOSUB ASCIHEX
PRINT A$; " ";
IF DAHLFLAG = 0 THEN GOTO RIGHT2
CHAR$ = MID$(INNY2$, 4, 1)
GOSUB ASCIHEX
PRINT A$; " ";
CHAR$ = MID$(INNY2$, 5, 1)
GOSUB ASCIHEX
PRINT A$; " ";
CHAR$ = MID$(INNY2$, 6, 1)
GOSUB ASCIHEX
PRINT A$; " ";
CHAR$ = MID$(INNY2$, 7, 1)
GOSUB ASCIHEX
PRINT A$; " ";
CHAR$ = RIGHT$(INNY2$, 1)
GOSUB ASCIHEX
PRINT A$; " ";
PRINT " BYTES="; LEN(INNY2$); " "
GOTO DAHLGREN
PRINT "FIRST RECORD A REFERENCE ANIMAL CALL, THEN"
PRINT "RECORD A TEST CALL. HAVE THESE TWO THINGS BEEN"
PRINT "BEEN COMPLETED? (Y/N)"
TAGVAR$ = INKEY$
IF TAGVAR$ = "N" OR TAGVAR$ = "n" THEN PRINT "THEN DO THEM NOW": PRINT " ": GOTO CAPTURE
IF TAGVAR$ = "Y" OR TAGVAR$ = "y" THEN GOTO CAPT2
GOTO IN95
PRINT "START TIME IS: "; TIMES
INNY$ = INPUT$(LOC(1), #1)
PRINT "HOST PC NOW AWAITING ANALYSIS COMPLETION"
PRINT "ESTIMATED TIME IS 17 MINUTES FROM RECORDING"
PRINT "OF TEST CALL"
TEMP1$ = " "
GETMORE95: INNY$ = INPUT$(LOC(1), #1)
IF INNY$ <> " " AND LEN(INNY$) > 4 THEN PRINT "ANALYSIS COMPLETED, WILL NOW REQUEST RESULTS": GOTO GETEM1
IF INNY$ <> " " THEN TEMP1$ = TEMP1$ + INNY$
IF LEN(TEMP1$) >= 6 THEN INNY$ = TEMP1$: GOTO GETEM1
GOSUB STALL
GOTO GETMORE95
PRINT CHR$(7): GOSUB STALL: PRINT CHR$(7): PRINT INNY$
GOSUB EVAL2
PRINT "MATCH PARAMETER IS: "; CALCADD
PRINT " ": PRINT "HIT THE SPACE BAR TO CONTINUE"

```

```

INSPACE1:  TAGVAR$ = INKEY$
            IF TAGVAR$ = " " THEN GOTO NEXTTEST1
            GOTO INSPACE1
NORMAL:    PRINT "YOU MUST PERFORM A FULL MULTI-FREQUENCY"
            PRINT "MULTI-EPOCH TEST FIRST. HAS THIS BEEN DONE YET? (Y/N) "
TA1:      TAGVAR$ = INKEY$
            IF TAGVAR$ = "Y" OR TAGVAR$ = "y" THEN GOTO NORM2
            IF TAGVAR$ = "N" OR TAGVAR$ = "n" THEN GOTO NEXTTEST1
            GOTO TA1
NORM2:    PRINT "DO YOU WISH TO WRITE THE DATA TO DISK? (Y/N) "
TA911:    TAGVAR$ = INKEY$
            IF TAGVAR$ = "Y" OR TAGVAR$ = "y" THEN WRITFLAG = 1: GOTO NORM3
            IF TAGVAR$ = "N" OR TAGVAR$ = "n" THEN WRITFLAG = 0: GOTO NORM3
            GOTO TA911
NORM3:    IF WRITFLAG = 1 THEN GOSUB INITDATE: OPEN FILENAME$ + ".DAT" FOR OUTPUT AS #3
            SUM = 0
            MIN = 256: MAX = 0
            FOR L = 1 TO 101
                EFLINE$ = "N"
                INNY$ = INPUT$(LOC(1), #1)
                PRINT #1, EFLINE$;
                FOR J = 1 TO 200
                    NEXT J
                INNY$ = INPUT$(LOC(1), #1)
                D = VAL(INNY$)
                IF D < 0 THEN D = D + 256
                IF L = 1 THEN PRINT "AVERAGE CALCULATED IN UP IS: "; D: GOTO CCB
                IF INT((L - 1) / 10) = (L - 1) / 10 THEN PRINT D
                IF WRITFLAG = 1 THEN WRITE #3, D
                IF INT((L - 1) / 10) <> (L - 1) / 10 THEN PRINT D;
                IF D < MIN THEN MIN = D
                IF D > MAX THEN MAX = D
                SUM = SUM + D
            NEXT L
            PRINT "SUM= "; SUM; " AVERAGE = "; SUM / 100
            PRINT "MAX= "; MAX; " MIN= "; MIN; " MAX:AVE= "; MAX / (SUM / 100)
            CLOSE #3
            TAGVAR$ = INKEY$
            IF TAGVAR$ = " " THEN GOTO NEXTTEST1
            GOTO TA2
TA2:      TAGVAR$ = INKEY$
            IF TAGVAR$ = " " THEN GOTO NEXTTEST1
            GOTO TA2
MULTTEST: CLS
            PRINT "TRANSFER FREQUENCY PATTERN TO DACIS PROCESSOR"
            PRINT "ENTER PEAK AMPLITUDE: "; INPUT AMP
            IF AMP > 127 THEN PRINT "AMPLITUDE TOO LARGE, TRY AGAIN": GOTO ENTAMP2
            PRINT "ENTER START FREQUENCY": INPUT STARTFREQ
            PRINT "ENTER END FREQ": INPUT ENDFREQ

```

```

FOR L = 1 TO 3072
PRINT L
A = INT(AMP * SIN(2 * 3.14159 * FREQ * (L / 6000)))
IF A < 0 THEN A = A + 256
PRINT #1, CHR$(A);
FOR K = 1 TO 3
NEXT K
NEXT L
PRINT "DOWNLOAD FINISHED, WILL NOW CHECK FOR RECEPTION"
EFLINE$ = "C"
PRINT #1, EFLINE$;
GOSUB STALL
RS232 = 99

GOTO STRTHERE:
PRINT "DO YOU WANT TO PRINT PARAMETERS? (Y/N)";
TAGVAR$ = INKEY$
IF TAGVAR$ = "Y" OR TAGVAR$ = "y" THEN PRNTFLAG = 1: PRINT "YES": LPRINT DATE$: GOTO WRITEFILE
IF TAGVAR$ = "N" OR TAGVAR$ = "n" THEN PRNTFLAG = 0: PRINT "NO": GOTO WRITEFILE
GOTO TRY822

WRITEFILE: PRINT "DO YOU WANT TO WRITE DATA TO DISK? (Y/N)";
TAGVAR$ = INKEY$
IF TAGVAR$ = "Y" OR TAGVAR$ = "y" THEN DISKFLAG = 1: PRINT "YES": GOTO AUDICONT
IF TAGVAR$ = "N" OR TAGVAR$ = "n" THEN DISKFLAG = 0: PRINT "NO": GOTO AUDICONT
GOTO TRY888

AUDICONT: ANALNUM = 1
PRINT "DO YOU WANT TO DO REPETITIVE AUTOMATIC ANALYSES? (Y/N)";
TAGVAR$ = INKEY$
IF TAGVAR$ = "Y" OR TAGVAR$ = "y" THEN AUTOFLAG = 1: PRINT "YES": GOTO STARTAUD
IF TAGVAR$ = "N" OR TAGVAR$ = "n" THEN AUTOFLAG = 0: PRINT "NO": GOTO STARTAUD
GOTO TRY420

STARTAUD: IF AUTOFLAG = 1 THEN GOTO GETFILT8
PRINT "HAS AN AUDIO CAPTURE BEEN PERFORMED? (Y/N) "
TAGVAR$ = INKEY$
IF TAGVAR$ = "Y" OR TAGVAR$ = "y" THEN GOTO GETFILT8
IF TAGVAR$ = "N" OR TAGVAR$ = "n" THEN PRINT "THEN DO ONE NOW": GOTO NEXTTEST1
GOTO IN829

MULTRECV: PRINT "DOWNLOAD FINISHED, WILL NOW CHECK FOR RECEPTION"
EFLINE$ = "C"
PRINT #1, EFLINE$;
GOSUB STALL
RS232 = 99
INNY$ = INPUT$(LOC(1), #1)
RS232 = 11
IF INNY$ = "X" THEN PRINT "DOWNLOAD SUCCESSFUL": GOTO GETFILT8
IF INNY$ <> "X" THEN PRINT "DOWNLOAD WAS NOT SUCCESSFUL"
DCNT = 1
GETFILT8:

```

```

IF PRNTFLAG = 1 THEN LPRINT "FREQ      "; "250 500 750 1000 1250 1500 1750 2000 2250 2500"
FOR JJ = 1 TO 10
SUMFILT(JJ) = 0
NEXT JJ
GETTIMFLG = 1
IF NORMFLAG = 1 THEN GOTO GETFILT9
GOTO GETFILT6:
EFLINES$ = "D"
RS232 = 99
PRINT #1, EFLINES$
RS232 = 11
FOR J = 1 TO 2000
FOR K = 1 TO 10
NEXT K
NEXT J
SUM1 = 0
GETFILT9: INNY$ = ""
INNY$ = INPUT$(LOC(1), #1)
IF LEN(INNY$) >= 12 AND RIGHT$(INNY$, 1) = " " THEN GOTO GOTSAMP
IF LEN(INNY$) <> 0 THEN INFIRST$ = INFIRST$ + INNY$
IF LEN(INFIRST$) >= 12 THEN INNY$ = INFIRST$: INFIRST$ = "": GOTO GOTSAMP
FOR R = 1 TO 500
NEXT R
GOTO GETFILT9
GOTSAMP: IF GETTIMFLG = 1 AND DISKFLAG = 1 THEN GOSUB INITDATE
IF GETTIMFLG = 1 THEN CLS : PRINT "DATE="; " DATE$; " START="; " TIME$; " ANALYSIS NUMBER="; ANALNUM: ANALNUM
= ANALNUM + 1
IF GETTIMFLG = 1 THEN PRINT "FREQ      "; "250 500 750 1000 1250 1500 1750 2000 2250 2500"
IF GETTIMFLG = 1 AND DISKFLAG = 1 THEN OPEN FILENAME$ + ".DAT" FOR OUTPUT AS #3
IF GETTIMFLG = 1 AND DISKFLAG = 1 THEN PRINT #3, "DATE="; " DATE$; " TIME="; " TIME$
IF DISKFLAG = 1 AND GETTIMFLG = 1 THEN PRINT #3, "FREQ      "; "250 500 750 1000 1250 1500 1750 2000 2250 2500"
2500": GETTIMFLG = 0
GOSUB EVAL1
INDEX1 = DCNT - 10 * INT(DCNT / 10)
N = INT((DCNT - 1) / 10) + 1: M = DCNT - 10 * (N - 1)
IF INT((DCNT - 1) / 10) = (DCNT - 1) / 10 THEN PRINT "EPOCH"; (DCNT + 9) / 10; " ";
IF INT((DCNT - 1) / 10) = (DCNT - 1) / 10 AND PRNTFLAG = 1 THEN LPRINT "EPOCH"; (DCNT + 9) / 10; " ";
IF INT((DCNT - 1) / 10) = (DCNT - 1) / 10 AND DISKFLAG = 1 THEN PRINT #3, "EPOCH"; (DCNT + 9) / 10; " ";
IF INT((DCNT - 1) / 10) = (DCNT - 1) / 10 AND DCNT <> 91 THEN PRINT " ";
A$ = STR$(INT(CALCADD / 100000)): ZZ = INT(CALCADD / 100000)
SUMFILT(INDEX1) = SUMFILT(INDEX1) + ZZ
MATRIX(N, M) = ZZ
SUM1 = SUM1 + ZZ
GOSUB MAKE4
GOTO PRINT104
IF LEN(A$) = 1 THEN A$ = A$ + " "

```

```

IF LEN(A$) = 2 THEN A$ = A$ + " "
IF LEN(A$) = 3 THEN A$ = A$ + " "
IF LEN(A$) = 4 THEN A$ = A$ + " "

```

```

RETURN

```

```

PRINT104: PRINT A$;

```

```

IF PRNTFLAG = 1 THEN LPRINT A$;
IF DISKFLAG = 1 THEN PRINT #3, A$;
DCNT = DCNT + 1

```

```

IF INT((DCNT - 1) / 10) = (DCNT - 1) / 10 THEN PRINT " "
IF INT((DCNT - 1) / 10) = (DCNT - 1) / 10 AND PRNTFLAG = 1 THEN LPRINT " "
IF INT((DCNT - 1) / 10) = (DCNT - 1) / 10 AND DISKFLAG = 1 THEN PRINT #3, " "

```

```

REM IF DCNT = 101 THEN GOSUB FINDATA

```

```

REM IF DCNT = 101 THEN PRINT "SUM="; SUM1; " AVERAGE="; SUM1 / 100: GOTO CHKSPACE
IF DCNT = 101 THEN GOTO CHKSPACE

```

```

GOTO GETFILT9

```

```

CHKSPACE: PRINT "NORMALIZED PARAMETERS"

```

```

IF PRNTFLAG = 1 THEN LPRINT "NORMALIZED PARAMETERS"

```

```

IF DISKFLAG = 1 THEN PRINT #3, "NORMALIZED PARAMETERS"

```

```

PRINT "FREQ "; "250 500 750 1000 1250 1500 1750 2000 2250 2500"

```

```

IF DISKFLAG = 1 THEN PRINT #3, "250 500 750 1000 1250 1500 1750 2000 2250 2500"

```

```

IF PRNTFLAG = 1 THEN LPRINT "FREQ "; "250 500 750 1000 1250 1500 1750 2000 2250 2500"

```

```

DCNT = 1: SUM = 0

```

```

INFIRST$ = ""

```

```

INNY$ = ""

```

```

INNY$ = INPUT$(LOC(1), #1)

```

```

IF LEN(INNY$) = 0 THEN GOTO GET1017

```

```

IF RIGHT$(INNY$, 1) <> " " THEN INFIRST$ = INFIRST$ + INNY$: INFIRST$ = ""

```

```

INFIRST$ = INFIRST$ + INNY$: A$ = INFIRST$: INFIRST$ = ""

```

```

IF LEFT$(A$, 1) = " " THEN GOTO REDO1019

```

```

IF LEFT$(A$, 1) = "-" THEN A$ = STR$(256 + VAL(A$))

```

```

IF LEFT$(A$, 1) = " " THEN A$ = RIGHT$(A$, LEN(A$) - 1)

```

```

IF LEFT$(A$, 1) = " " THEN A$ = RIGHT$(A$, LEN(A$) - 1)

```

```

SUM = SUM + VAL(A$)

```

```

GOSUB MAKE4

```

```

IF INT((DCNT - 1) / 10) = (DCNT - 1) / 10 THEN PRINT "EPOCH"; (DCNT + 9) / 10; " ";

```

```

IF INT((DCNT - 1) / 10) = (DCNT - 1) / 10 AND PRNTFLAG = 1 THEN LPRINT "EPOCH"; (DCNT + 9) / 10; " ";

```

```

IF INT((DCNT - 1) / 10) = (DCNT - 1) / 10 AND DISKFLAG = 1 THEN PRINT #3, "EPOCH"; (DCNT + 9) / 10; " ";

```

```

IF INT((DCNT - 1) / 10) = (DCNT - 1) / 10 AND DCNT <> 91 THEN PRINT " ";

```

```

IF INT(DCNT / 10) = DCNT / 10 THEN PRINT A$

```

```

IF INT(DCNT / 10) = DCNT / 10 AND PRNTFLAG = 1 THEN LPRINT A$

```

```

IF INT(DCNT / 10) = DCNT / 10 AND DISKFLAG = 1 THEN PRINT #3, A$

```

```

IF INT(DCNT / 10) <> DCNT / 10 THEN PRINT A$;

```

```

IF INT(DCNT / 10) <> DCNT / 10 AND PRNTFLAG = 1 THEN LPRINT A$;

```

```

IF INT(DCNT / 10) <> DCNT / 10 AND DISKFLAG = 1 THEN PRINT #3, A$;

```

```

DCNT = DCNT + 1

```

```

IF DCNT = 101 THEN GOTO IN1017

```

```

IN1017:      GOTO GET1017
PRINT "AVERAGE IS: "; SUM / 100;
IF PRNTFLAG = 1 THEN LPRINT "AVERAGE IS: "; SUM / 100
IF DISKFLAG = 1 THEN PRINT #3, "AVERAGE IS: "; SUM / 100
CLOSE #3
IF AUTOFLAG = 1 THEN GOTO GETFILT8
TAGVAR$ = INKEY$
IF TAGVAR$ = " " THEN GOTO NEXTTEST1
GOTO IN10172
STRTHERE:   INNY$ = INPUT$(LOC(1), #1)
IF INNY$ = "X" THEN PRINT "DOWNLOAD SUCCESSFUL"
IF INNY$ <> "X" THEN PRINT "DOWNLOAD NOT SUCCESSFUL"; PRINT INNY$, LEN(INNY$)
NEXTFREQ:   PRINT "AMPLITUDE= "; AMP; " INPUT FREQUENCY= "; FREQ
IF WRITDISK = 1 THEN PRINT #2, "AMPLITUDE= "; AMP; " INPUT FREQUENCY= "; FREQ
RS232 = 11
EFLINES$ = "D": SECONDS = 3600 * VAL(LEFT$(TIME$, 2)) + 60 * VAL(MID$(TIME$, 4, 2)) + VAL(RIGHT$(TIME$, 2))
PRINT #1, EFLINES$;
FOR J = 1 TO 3000
FOR K = 1 TO 10
NEXT K
NEXT J
RS232 = 99
FREQREC = 0
GETFILT:    INNY$ = " "
GETFILT2:   INNY$ = INPUT$(LOC(1), #1)
'12=MINIMUM TRANSMISSION PER FILTER OUTPUT
IF LEN(INNY$) >= 12 AND RIGHT$(INNY$, 1) = " " THEN GOTO PRINTDATA
IF LEN(INNY$) <> 0 THEN INFIRST$ = INFIRST$ + INNY$
IF LEN(INFIRST$) >= 12 THEN INNY$ = INFIRST$: INFIRST$ = "": GOTO PRINTDATA
FOR L = 1 TO 500
NEXT L
GOTO GETFILT2
PRINTDATA:  GOSUB EVAL1
FILTTIME = 3600 * VAL(LEFT$(TIME$, 2)) + 60 * VAL(MID$(TIME$, 4, 2)) + VAL(RIGHT$(TIME$, 2)) - SECONDS
PRINT "TIME TO COMPLETE FILTER= "; INT(FILTTIME / 60); " "; FILTTIME - 60 * INT(FILTTIME / 60); " ";
IF WRITDISK = 1 THEN PRINT #2, "TIME TO COMPLETE FILTER= "; INT(FILTTIME / 60); " "; FILTTIME - 60 *
INT(FILTTIME / 60); " ";
FREQREC = FREQREC + 1: ACTFREQ = 250 * FREQREC
PRINT "FREQ="; ACTFREQ; "Hz OUTPUT= "; CALCADD
IF WRITDISK = 1 THEN PRINT #2, "FREQ="; ACTFREQ; "Hz OUTPUT= "; CALCADD
IF DOALL = 1 AND FREQREC = 10 THEN GOTO CHECKALL
IF FREQREC = 10 AND INKEY$ = " " THEN CLOSE #2: GOTO NEXTTEST1
IF FREQREC <> 10 THEN GOTO GETFILT2
GOTO WAIT1
CHECKALL:   IF FREQ = 2500 THEN GOTO NEXTTEST1
PRINT #2, " "

```

```

FREQ = FREQ + 250: GOTO DUMPSINE
MEGATEST: EFLINES$ = "B"
CLS
MEGATEST2: PRINT "THIS TEST WILL SUM ALL INTEGERS BETWEEN THE START"
PRINT " " AND ENDING VALUES AND RETURN THE RESULT"
PRINT "ENTER START VALUE: ": INPUT S1$: S1 = VAL(S1$)
PRINT "ENTER END VALUE: ": INPUT S2$: S2 = VAL(S2$)
RESULT = (S1 + S2) * (ABS(S2 - S1) + 1) / 2
IF S1 < S2 THEN LS1 = S1
IF S1 < S2 THEN GS1 = S2
IF S2 < S1 THEN LS1 = S2
IF S2 < S1 THEN GS1 = S1
IF LS1 >= 0 THEN LESSER1 = INT(LS1 / 256)
IF LS1 >= 0 THEN LESSER2 = LS1 - 256 * LESSER1
IF GS1 >= 0 THEN GREATER1 = INT(GS1 / 256)
IF GS1 >= 0 THEN GREATER2 = GS1 - 256 * GREATER1
IF LS1 < 0 THEN LESSER1 = 128 + INT((LS1 + 32768) / 256)
IF LS1 < 0 THEN LESSER2 = (LS1 + 32768) - 256 * (LESSER1 - 128)
IF GS1 < 0 THEN GREATER1 = 128 + INT((GS1 + 32768) / 256)
IF GS1 < 0 THEN GREATER2 = (GS1 + 32768) - 256 * (GREATER1 - 128)
INNY$ = INPUT$(LOC(1), #1)
PRINT #1, EFLINES$
FOR DELAY = 1 TO 400
NEXT DELAY
PRINT #1, CHR$(LESSER1)
PRINT #1, CHR$(LESSER2)
PRINT #1, CHR$(GREATER1)
PRINT #1, CHR$(GREATER2)
GOSUB STALL
RS232 = 99
INNY$ = INPUT$(LOC(1), #1)
PRINT INNY$, LEN(INNY$)
RS232 = 11
GOSUB EVAL1
GOTO DONE820
EVAL2: FOR L = 1 TO 5
IF MID$(INNY$, L, 1) = " " THEN LOHIBYTE = VAL(LEFT$(INNY$, L)): GOTO NEXT991
NEXT L
NEXT991: LOLOBYTE = VAL(RIGHT$(INNY$, LEN(INNY$) - L))
A = 256 * LOHIBYTE + LOLOBYTE
IF LOHIBYTE >= 0 AND LOLOBYTE < 0 THEN A = 256 * LOHIBYTE + (LOLOBYTE - 256)
IF LOHIBYTE < 0 AND LOLOBYTE < 0 THEN A = 256 * (LOHIBYTE + 256) + (LOLOBYTE + 256)
IF LOHIBYTE < 0 AND LOLOBYTE >= 0 THEN A = 256 * LOHIBYTE + (LOLOBYTE - 256)
CALCADD = A
RETURN
EVAL1: FOR L = 1 TO 5

```



```

IF MID$(INNY$, L, 1) = " " THEN LOHIBYTE = VAL(LEFT$(INNY$, L)): GOTO NEXT891
NEXT L
NEXT891: IF LEFT$(INNY$, 1) = "0" AND MID$(INNY$, 2, 1) = "0" THEN LOHIBYTE = 0
FOR K = L + 1 TO L + 5
IF MID$(INNY$, K, 1) = " " THEN LOLOBYTE = VAL(MID$(INNY$, L + 1, K - L)): GOTO NEXT892
NEXT K
NEXT892: IF MID$(INNY$, L + 1, 1) = "0" AND MID$(INNY$, L + 2, 1) = "0" THEN LOLOBYTE = 0
FOR L = K + 1 TO K + 5
IF MID$(INNY$, L, 1) = " " THEN HIHIBYTE = VAL(MID$(INNY$, K + 1, L - K)): GOTO NEXT81X
NEXT L
NEXT81X: IF MID$(INNY$, K + 1, 1) = "0" AND MID$(INNY$, K + 2, 1) = "0" THEN HIHIBYTE = 0
HILOBYTE = VAL(RIGHT$(INNY$, LEN(INNY$) - L))
GOTO NOPRINT
NOPRINT: PRINT INNY$: " "; LOHIBYTE; " "; HIHIBYTE; " "; HILOBYTE
A = 256 * LOHIBYTE + LOLOBYTE
B = 256 * HIHIBYTE + HILOBYTE
IF LOHIBYTE >= 0 AND LOLOBYTE < 0 THEN A = LOHIBYTE * 256 + (LOLOBYTE + 256)
IF HIHIBYTE >= 0 AND HILOBYTE < 0 THEN B = HIHIBYTE * 256 + (HILOBYTE + 256)
IF LOHIBYTE < 0 AND LOLOBYTE < 0 THEN A = ((LOHIBYTE + 256) * 256) + (LOLOBYTE + 256) - 65536
IF HIHIBYTE < 0 AND HILOBYTE < 0 THEN B = ((HIHIBYTE + 256) * 256) + (HILOBYTE + 256) - 65536
IF LOHIBYTE < 0 AND LOLOBYTE >= 0 THEN A = ((LOHIBYTE + 256) * 256) + LOLOBYTE - 65536
IF HIHIBYTE < 0 AND HILOBYTE >= 0 THEN B = ((HIHIBYTE + 256) * 256) + HILOBYTE - 65536
CALCADD = B * 32768 + A
RETURN
DONE820: PRINT "A="; A; "B="; B
PRINT "CALCULATED SUM="; RESULT
PRINT "PROCESSOR SUM="; CALCADD
GOTO NEXTTEST1
MULTIPLY: PRINT "ENTER VALUE 1=-128 TO +127"
INPUT A1
IF A1 < -128 OR A1 > 127 THEN GOTO MULTIPLY
X1 = A1
IF A1 < 0 THEN A1 = A1 + 256
PRINT A1, STR$(A1), LEN(STR$(A1)), RIGHT$(STR$(A1), 2), VAL(STR$(A1))
PRINT "ENTER SECOND VALUE =-128 TO +127"
INPUT A2
IF A2 < -128 OR A2 > 127 THEN GOTO A2AGAIN
X2 = A2
IF A2 < 0 THEN A2 = A2 + 256
INNY$ = INPUT$(LOC(1), #1)
EFLINE$ = "M"
PRINT #1, EFLINE$
FOR DELAY = 1 TO 300
NEXT DELAY
PRINT #1, CHR$(A1)
FOR DELAY = 1 TO 300

```

```

NEXT DELAY
PRINT #1, CHR$(A2)
FOR DELAY = 1 TO 300
NEXT DELAY
RS232 = 99
INNY$ = INPUT$(LOC(1), #1)
RS232 = 11
FOR L = 1 TO 4
IF MID$(INNY$, L, 1) = " " THEN HIGHBYTE = VAL(LEFT$(INNY$, LEN(INNY$) - L))
NEXT L
IF LEFT$(INNY$, 1) = "0" AND MID$(INNY$, 2, 1) = "0" THEN HIGHBYTE = 0
FOR L = 1 TO 8
IF MID$(INNY$, L, 1) = " " THEN LOWBYTE = VAL(RIGHT$(INNY$, LEN(INNY$) - L)): GOTO PRINTMUL
NEXT L
PRINTMUL: IF LOWBYTE < 0 THEN LOWBYTE = LOWBYTE + 256
PRINT INNY$, HIGHBYTE, LOWBYTE
PRINT "MULTIPLIED RESULT IS: "; HIGHBYTE * 256 + LOWBYTE
PRINT "EXPECTED RESULT IS: "; X1 * X2
GOTO NEXTTEST1

DUMPWGHT: INNY$ = INPUT$(LOC(1), #1)
FOR L = 1 TO 10
LINER$ = "FILTER"
IF L < 10 THEN LINER$ = "FILTER "
PRINT LINER$; L; " ";
FOR K = 1 TO 23
EFLINE$ = "F"
PRINT #1, EFLINE$
FOR DELAY = 1 TO 300
NEXT DELAY
RS232 = 99
INNY$ = INPUT$(LOC(1), #1)
RS232 = 11
IF LEN(INNY$) = 3 THEN INNY$ = " " + INNY$
IF LEN(INNY$) = 2 THEN INNY$ = " " + INNY$
IF LEN(INNY$) = 1 THEN INNY$ = " " + INNY$
PRINT INNY$;
IF K <> 11 THEN PRINT " ";
NEXT K
PRINT " "
NEXT L
GOTO NEXTTEST1

SPECTRUM: INNY$ = INPUT$(LOC(1), #1)
PRINT "FREQUENCY    POWER"
FOR L = 1 TO 10
FREQ = 250 * L
PRINT " "; FREQ; " ";

```

```

EFLINES = "O"
PRINT #1, EFLINES
FOR DELAY = 1 TO 300
NEXT DELAY
RS232 = 99
INNY$ = INPUT$(LOC(1), #1)
RS232 = 11
POWER = VAL(INNY$)
IF POWER < 0 THEN POWER = POWER + 256
PRINT POWER
NEXT L
GOTO NEXTTEST1

WATERFALL: PRINT "FREQ. EPOCH1 EPOCH2 EPOCH3 EPOCH4 EPOCH5 EPOCH6 EPOCH7 EPOCH8 EPOCH9 EPOCH10"
FOR L = 1 TO 10
FREQ = 250 * L
A$ = STR$(FREQ)
IF LEN(A$) < 5 THEN A$ = A$ + " "
PRINT A$; " ";
EFLINES = "W"
FOR J = 1 TO 10
PRINT #1, EFLINES
FOR DELAY = 1 TO 300
NEXT DELAY
RS232 = 99
INNY$ = INPUT$(LOC(1), #1)
RS232 = 11
POWER = VAL(INNY$)
IF POWER < 0 THEN POWER = POWER + 256
POW$ = STR$(POWER)
IF LEN(POW$) = 3 THEN POW$ = POW$ + " "
IF LEN(POW$) = 4 THEN POW$ = POW$ + " "
IF LEN(POW$) = 2 THEN POW$ = POW$ + " "
IF LEN(POW$) = 1 THEN POW$ = POW$ + " "
PRINT POW$; " ";
NEXT J
PRINT " "
NEXT L
GOTO NEXTTEST1

FINDATA: PRINT "AVERAGE ";
FOR M = 1 TO 10
SUMROW = 0
FOR N = 1 TO 10
SUMROW = SUMROW + MATRIX(N, M)
NEXT N
A$ = STR$(INT(SUMROW / 10))
GOSUB MAKE4

```

```

PRINT A$;
NEXT M
PRINT " "
PRINT "ABSDEV "
FOR M = 1 TO 10
  ABSDEV = 0
  FOR N = 1 TO 10
    SUMROW = SUMROW + MATRIX(N, M)
  NEXT N
  ROWAVE = SUMROW / 10
  FOR N = 1 TO 10
    ABSDEV = ABSDEV + ABS(MATRIX(N, M) - ROWAVE)
  NEXT N
  A$ = STR$(INT(ABSDEV / 10))
  GOSUB MAKE4
  PRINT A$;
NEXT M
PRINT " "
RETURN
FOR L = 0 TO 255
  IF CHR$(L) = CHAR$ THEN VALUE = L: A$ = STR$(VALUE): RETURN
NEXT L
RETURN
FOR L = 0 TO 255
  IF CHR$(L) = CHAR$ THEN VALUE = L: GOTO CONVERTH
NEXT L
RETURN
CONVERTH: A = INT(VALUE / 16)
IF A > 9 THEN GOSUB CONVERTH1: GOTO SKIPIT1
C$ = STR$(A)
A$ = C$:
B = VALUE - (16 * A)
IF B > 9 THEN GOSUB CONVERTH1: GOTO SKIPIT2
C$ = STR$(B)
A$ = A$ + RIGHT$(C$, 1) + "H"
RETURN
CONVERTH1: IF A = 10 THEN C$ = "A"
IF A = 11 THEN C$ = "B"
IF A = 12 THEN C$ = "C"
IF A = 13 THEN C$ = "D"
IF A = 14 THEN C$ = "E"
IF A = 15 THEN C$ = "F"
RETURN
PRINT "PROGRAM ABORTED"
END
RETURN

```

Enclosure 4
Filter Output Data and Analysis of Wolf Calls

COMPARISON MATRIX

8/21/98

HOWL COMPOSITE										
	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	126	116	85	29.5	21	18	16.75	6.5	12	9.25
EPOCH 2	132	120.5	82.5	39.5	19.75	17.75	8.5	7.75	12.25	10.25
EPOCH 3	113	127.5	91.5	34	22	18.25	11.25	11	12.75	9.25
EPOCH 4	118.75	118.75	74.75	32.5	19.5	17.25	9.5	8.25	12.75	9
EPOCH 5	122.75	109.25	71.5	29.5	18.25	16.75	10	5.75	12	9.25
EPOCH 6	129.75	109.25	65.5	25.75	20.25	17.75	14.25	6.5	12.75	9.25
EPOCH 7	100.25	96	68.75	25.75	19	17.75	8	6.25	12.25	8.5
EPOCH 8	85.25	85.5	63.25	25.25	18.5	16	6.25	6.25	12.25	7
EPOCH 9	74.5	63.25	42.75	23.5	16.5	15	5.5	5.25	12	5
EPOCH 10	82.75	61.75	48.5	27	16.5	15	4.5	4.25	12	6
SUM	1085	1007.75	694	292.25	191.25	169.5	94.5	67.75	123	82.75
STDV	21.26	23.54	15.41	4.91	1.78	1.22	3.85	1.87	0.33	1.69
GROWL COMPOSITE										
	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	107.5	91.75	54	28	27	24.5	29.25	22.5	17.5	9.25
EPOCH 2	116.25	94.25	67.5	33.5	30.25	28.75	35.5	24.75	19.75	11.5
EPOCH 3	126.75	94.75	57	30.75	28.5	23	20.25	13.5	17.25	9.75
EPOCH 4	103.5	82.5	49	28	27.25	23.5	26.5	16.75	16.5	9.5
EPOCH 5	97.75	84.25	57.25	28.75	25	24	23.25	19.5	18.5	9.5
EPOCH 6	102	71.5	47.25	28.25	26.25	20.75	14.75	12	15.75	8.25
EPOCH 7	90.25	50.75	38.25	25.5	22	19.25	14.5	10	15.5	7.5
EPOCH 8	83.25	90.75	55.25	32.25	25.75	21.25	18.75	13.25	18.25	12.75
EPOCH 9	125.75	79.75	46.75	28.5	27.75	20.25	19	14	15.75	8.75
EPOCH 10	101	73.75	51.5	28.75	25.25	22.75	21	17.5	17	8.75
SUM	1054	814	523.75	292.25	265	228	222.75	163.75	171.75	95.5
STDV	14.16	13.52	7.85	2.32	2.24	2.71	6.57	4.74	1.37	1.54
BARK COMPOSITE										
	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	62	59.25	61.25	48	61.5	39.75	48.75	36.5	20	13.25
EPOCH 2	63.25	62.5	47.5	40	45.75	28.75	42	30	17	9.75
EPOCH 3	57.5	58.5	50.5	47	50.5	30	40	26	16	10.5
EPOCH 4	64.5	77.5	51	54.25	47.5	29.75	42.75	27.75	17.5	11.5
EPOCH 5	53	66.25	44.25	56	42.75	23.5	26.75	18.75	13.25	8
EPOCH 6	48.25	53.75	49.75	48.5	50	27.25	30.75	22.5	15	9.75
EPOCH 7	47.5	56.75	36.25	35.75	35	22.5	26.75	20.25	13.75	7.75
EPOCH 8	49.75	61.75	51.5	45.5	48	30.5	36.5	31.75	17.75	11
EPOCH 9	51.25	73.25	44.25	37.75	36.5	23.75	28.25	22	13.5	8
EPOCH 10	67.25	94.25	51	43.75	42	27.75	32.5	26	13.25	9.75
SUM	564.25	663.75	487.25	456.5	459.5	283.5	355	261.5	157	99.25
STDV	7.38	12.24	6.47	6.60	7.61	4.94	25.00	5.54	2.33	1.74
WHINE COMPOSITE										
	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	70	102.25	36.25	40.5	26	39.5	37.5	13.75	16	10.25
EPOCH 2	74	106.5	33.75	42.5	29.5	41	63	129	22	13
EPOCH 3	63.75	30	19.75	20.75	15.75	14.75	7.5	6.75	13	6.25
EPOCH 4	65.25	55.25	27.25	24	18.75	20.75	17.25	8.75	14.75	7
EPOCH 5	91.25	93.75	44.5	32	28	39.25	47.5	22.75	17	11
EPOCH 6	171	166	77.5	46.75	54.75	53.75	54	55.5	26.5	22.25
EPOCH 7	69.75	41.5	25.75	34.25	29	25	25.75	17	19.25	11
EPOCH 8	64.5	33.25	24	22	17	15.25	9.75	8.25	13	8
EPOCH 9	52	26.75	21.75	23.25	18.25	36.25	15.5	16.75	30	20.55
EPOCH 10	75.25	37.5	21	22	20	18.75	14.75	15	19.25	22.75
SUM	796.75	692.75	331.5	308	257	304.25	292.5	293.5	190.75	317
STDV	33.63	46.01	17.44	9.76	11.48	13.27	25.00	37.71	5.67	61.34

COMPARISON MATRIX

8/21/98

COMPARISON MATRIX				
	HOWL COMP.	GROWL COMP.	BARK COMP.	WHINE COMP.
HOWL1	13.74056494	21.24445516	33.45437191	38.40712824
HOWL2	16.80887043	22.16059284	35.22939114	39.9940308
HOWL3	14.69347219	23.91959187	35.05909297	40.15504327
HOWL4	24.86761197	26.65148917	31.79158694	43.17218433
GROWL1	29.80181412	19.85414	29.29334395	42.67718946
GROWL2	18.01577434	14.64980802	27.15059852	30.23024148
GROWL3	22.19488961	14.35502961	24.54251413	37.0136799
GROWL4	17.44485956	16.59975527	24.74772717	36.97516058
BARK1	31.91814727	25.19080934	11.99979166	33.94521321
BARK2	32.46295004	26.74905372	19.02971361	41.93599289
BARK3	32.54901112	25.52580802	13.86001443	39.13537402
BARK4	33.95383815	26.61506857	17.09312727	34.49213679
WHINE1	36.71986009	33.99568906	35.50880173	2.739981752
WHINE2	36.41899401	33.21719246	34.08364711	1.103403825
WHINE3	36.21716616	32.8367458	33.29391836	2.225421308
WHINE4	35.71460661	32.577168	33.57655432	1.724093965
THRESHOLD				
17				
	% POS CORR.	FALSE NEG	%NEG CORR.	FALSE POS
	75	25	100	0
TOTAL CORRECT RATE				
	93.75			

AVERAGE IS: 38.21 08-29-1997

FREQ	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	651	648	757	169	139	106	52	50	69	66
EPOCH 2	621	879	904	175	120	112	49	47	69	79
EPOCH 3	543	816	600	174	126	98	59	41	72	51
EPOCH 4	596	617	356	232	109	91	59	39	69	-9396
EPOCH 5	702	639	240	233	107	87	41	34	69	50
EPOCH 6	686	692	245	137	100	85	50	35	71	42
EPOCH 7	605	595	241	129	100	85	51	28	69	45
EPOCH 8	7877	397	148	117	94	81	28	23	68	37
EPOCH 9	451	261	115	112	87	80	26	22	68	29
EPOCH 10	338	195	107	120	86	80	25	22	68	26

NORMALIZED PARAMETERS

FREQ	250	500	750	1000	1250	1750	2000	2250	2500	
EPOCH 1	121	121	140	30	25	19	09	08	12	12
EPOCH 2	115	164	167	31	22	20	08	08	12	14
EPOCH 3	100	152	111	31	23	18	11	07	13	08
EPOCH 4	110	115	66	42	19	15	11	06	12	08
EPOCH 5	131	118	44	42	19	15	07	06	12	08
EPOCH 6	127	128	45	25	18	15	08	06	12	07
EPOCH 7	112	110	44	23	18	15	08	04	12	07
EPOCH 8	100	73	26	20	17	14	04	03	12	06
EPOCH 9	83	47	20	20	15	14	04	03	12	04
EPOCH 10	66	35	19	22	15	14	03	03	12	04

08-29-1997

FREQ	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	955	636	232	149	100	97	44	31	70	57
EPOCH 2	923	686	301	169	109	103	49	39	72	59
EPOCH 3	877	868	642	201	136	119	107	74	73	68
EPOCH 4	947	852	589	209	128	130	91	72	70	75
EPOCH 5	1107	932	523	185	111	125	125	51	71	79
EPOCH 6	1045	767	379	174	125	142	250	63	86	87
EPOCH 7	629	468	282	162	122	122	97	71	74	53
EPOCH 8	421	493	526	164	117	97	53	81	68	56
EPOCH 9	342	390	420	145	95	89	46	64	68	45
EPOCH 10	332	348	651	127	98	84	38	35	68	52

NORMALIZED PARAMETERS

FREQ	250	500	750	1000	1250	1750	2000	2250	2500	
EPOCH 1	145	97	35	22	15	14	06	04	10	08
EPOCH 2	140	104	46	25	16	15	07	06	11	09
EPOCH 3	133	132	98	30	20	18	16	11	11	10
EPOCH 4	144	130	89	31	19	19	13	11	10	11
EPOCH 5	169	142	79	28	16	19	19	07	10	12
EPOCH 6	159	117	57	26	19	21	38	09	13	13
EPOCH 7	96	71	43	24	18	18	14	10	11	08
EPOCH 8	64	75	80	25	17	14	08	12	10	08
EPOCH 9	52	59	64	22	14	13	07	09	10	06
EPOCH 10	50	53	99	19	14	13	05	05	10	07

9/5

~~08-29~~-1997GROWL
at 650

FREQ	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	519	400	170	131	116	129	184	87	80	51
EPOCH 2	660	604	369	193	175	204	371	223	126	90
EPOCH 3	736	561	329	157	159	142	136	104	92	58
EPOCH 4	542	370	160	118	104	94	62	58	84	40
EPOCH 5	632	530	368	166	143	149	164	140	118	61
EPOCH 6	834	677	370	188	185	124	105	69	93	71
EPOCH 7	322	167	105	110	87	80	32	24	68	27
EPOCH 8	360	306	198	121	97	84	38	33	69	31
EPOCH 9	421	386	256	130	132	98	96	87	77	35
EPOCH 10	457	378	278	127	124	90	63	56	70	34

NORMALIZED PARAMETERS

FREQ	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	103	79	34	24	22	24	36	17	15	09
EPOCH 2	131	120	73	38	34	40	73	44	24	17
EPOCH 3	146	111	65	31	31	27	26	19	18	10
EPOCH 4	107	73	31	23	19	18	11	10	15	07
EPOCH 5	125	106	73	32	27	28	32	27	23	11
EPOCH 6	166	134	73	36	36	23	20	13	18	13
EPOCH 7	64	32	20	20	17	15	05	03	13	05
EPOCH 8	72	60	39	23	18	15	06	06	13	05
EPOCH 9	33	77	51	24	26	18	18	17	14	06
EPOCH 10	90	74	55	24	23	17	11	10	13	06

AVERAGE IS: 38.79

9/2
00-29-1997

GROW~
at 660
ON TAPE

FREQ	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	496	512	365	151	149	140	183	167	104	63
EPOCH 2	351	195	184	127	106	94	53	42	71	35
EPOCH 3	447	239	196	130	101	98	93	65	87	56
EPOCH 4	493	293	199	137	128	125	124	89	91	60
EPOCH 5	327	159	104	109	90	80	38	27	68	25
EPOCH 6	328	159	103	110	86	80	38	33	69	28
EPOCH 7	441	185	134	122	105	101	5	58	77	37
EPOCH 8	<u>1880</u>	<u>634</u>	334	211	152	117	93	78	112	120
EPOCH 9	<u>1014</u>	<u>375</u>	195	156	118	101	69	52	84	71
EPOCH 10	577	278	182	154	126	106	73	59	83	48

NORMALIZED PARAMETERS

FREQ	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	118	123	87	36	34	33	44	39	23	14
EPOCH 2	83	45	44	30	25	22	12	09	15	07
EPOCH 3	107	57	45	30	23	23	22	14	20	12
EPOCH 4	118	69	47	31	30	30	30	20	20	14
EPOCH 5	79	38	23	25	20	19	07	06	15	06
EPOCH 6	79	38	23	25	20	19	07	07	15	06
EPOCH 7	106	44	31	28	25	23	19	12	17	07
EPOCH 8	<u>49</u>	<u>152</u>	80	50	35	26	22	17	26	28
EPOCH 9	243	90	45	36	28	23	15	11	19	15
EPOCH 10	139	66	42	36	30	25	17	14	19	11

AVERAGE IS: 38.04

err.

GROWL
SAMPLE#3

09-03-1997

FREQ	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	543	386	172	119	104	100	74	65	87	44
EPOCH 2	749	627	454	198	188	175	201	163	130	79
EPOCH 3	726	573	281	156	140	105	70	83	84	56
EPOCH 4	323	170	106	110	9347	80	31	25	68	28
EPOCH 5	377	367	235	127	101	86	41	35	71	33
EPOCH 6	442	372	257	130	137	104	107	95	85	41
EPOCH 7	459	383	289	130	128	95	74	65	75	39
EPOCH 8	511	433	291	141	137	102	101	69	75	37
EPOCH 9	539	603	365	157	197	119	192	124	79	49
EPOCH 10	528	622	452	164	153	165	257	208	115	71

NORMALIZED PARAMETERS

FREQ	250	500	750	1000	1250	1500	1750	2000	2250	2500
EPOCH 1	103	74	32	22	19	19	13	12	16	07
EPOCH 2	144	120	87	38	35	32	38	30	24	15
EPOCH 3	139	110	53	29	26	20	12	08	15	10
EPOCH 4	62	32	20	20	16	15	05	03	12	05
EPOCH 5	72	70	44	24	19	16	07	06	12	06
EPOCH 6	84	70	49	24	25	20	20	19	15	07
EPOCH 7	88	73	55	24	24	17	13	12	13	06
EPOCH 8	99	83	55	26	25	19	19	12	13	06
EPOCH 9	103	116	69	30	38	22	36	22	15	08
EPOCH 10	101	119	87	31	29	31	49	39	21	12

AVERAGE IS: 37.6

GROWL
SAMPLE 4
≈ 660 on
TAPE

EPOCH 9	522	09-03-1997									
FREQ	250	500	750	1000	1250	1500	1750	2000	2250	2500	
EPOCH 1	461	398	281	131	145	101	110	100	74	37	
EPOCH 2	468	403	291	128	119	94	91	75	72	36	
EPOCH 3	501	445	282	144	151	104	91	62	73	36	
EPOCH 4	554	676	427	167	191	138	262	151	85	58	
EPOCH 5	502	535	391	152	153	145	205	171	109	65	
EPOCH 6	347	195	191	126	107	96	57	39	71	35	
EPOCH 7	448	240	206	136	101	98	97	64	87	54	
EPOCH 8	494	296	207	134	-530	117	128	82	92	57	
EPOCH 9	322	159	104	109	90	80	37	26	68	26	
EPOCH 10	326	158	101	109	86	80	38	33	69	28	

NORMALIZED PARAMETERS

FREQ	250	500	750	1000	1250	1500	1750	2000	2250	2500	
EPOCH 1	106	91	63	30	33	22	24	22	16	07	
EPOCH 2	107	92	66	28	27	21	19	16	16	07	
EPOCH 3	115	101	65	33	34	22	21	13	16	07	
EPOCH 4	127	156	98	38	44	31	60	34	19	12	
EPOCH 5	115	123	89	34	34	33	47	39	24	15	
EPOCH 6	79	44	44	28	24	21	12	09	15	07	
EPOCH 7	103	54	47	30	22	22	21	13	19	12	
EPOCH 8	114	68	47	30	24	25	28	18	21	12	
EPOCH 9	74	36	22	24	19	18	07	06	15	06	
EPOCH 10	74	36	22	24	19	18	07	07	15	06	

AVERAGE IS: 38.8

Enclosure 5
Article Presented at NASA Technology 2007
Conference

MINIATURE LOW POWER DIGITAL AUDIO CAPTURE AND IDENTIFICATION SYSTEM

**Protagoras N. Cutchis
Johns Hopkins University Applied Physics Laboratory
Johns Hopkins Road, Laurel, MD 20723
301-953-5013**

**William S. Seegar
U.S. Army Edgewood Research, Development and Engineering Center
Chemical and Biological Defense Command
Aberdeen Proving Ground
Aberdeen, MD 21010
410-671-2586**

We report here on the development of a small low power Digital Audio Capture and Identification System (DACIS) for the automatic identification of animal calls supported by the Strategic Environmental Research and Development Program (SERDP). The system is light enough and low power enough to be attached to larger birds so that the bird can act as a sentinel animal for its own environment. The system is designed to be used with a Global Positioning System (GPS) and Argos Platform Transmitter Terminal (PTT) which will allow accurate location fixes and telemetry of data via satellite. After capture and identification, the time and location of identified sounds are periodically transmitted up to the Argos satellite for relay to the earth.

INTRODUCTION

In 1981 the Bird-borne program was initiated at the Johns Hopkins University, Applied Physics Laboratory (APL) under U.S. Army funding, to develop a capability to locate (i.e., track) and monitor small highly mobile organisms on a local, regional and global scale. Initially, the program focused on the development of small Platform Transmitter Terminals (PTTs) to be tracked via the French-U.S. Argos satellite system. Since the inception of the program, miniaturization has led to the fielding of transmitters that weigh less than one ounce and are capable of interfacing with an array of sensors.

The primary objective of the Bird-borne and the Remote Environmental Sensing Technology (REST) program (follow-up initiated in 1991) has been to develop a system for the remote tracking and monitoring of free ranging organisms that pose especially difficult field problems for study. Biotelemetry can be used to collect information from the environment surrounding the animal (temperature, humidity, altitude) as well as behavioral and physiological parameters (motion, core temperature and heart rate) of the organism. Biotelemetry has enabled scientists to accurately study behavior, home range, and habitat use of wildlife for basic research and the development of management plans for conservation.

However, for studies of free ranging organisms that travel long distances over extended periods and frequent inaccessible habitats due to geographic or boundary restrictions, such as

military installations, space-based tracking and monitoring systems are advantageous. Remote tracking and monitoring systems can support effective study of these organisms and aid in identifying their range and critical habitat requirements for breeding, migration and wintering.

The planned uses for the acoustic system are to identify the prey on which predatory animals are feeding and identification of known calls of the animal on which the unit is mounted to identify behaviors correlated with those sounds.. While the identification of prey is a desirable goal, it was felt that the number of targets for identification would be too large for a proof of concept. Therefore, the present system is focusing on the latter.

The DOD has established requirements for environmental research, technology development and land management and supports a variety of programs such as Legacy and the Strategic Environmental Research and Development Program (SERDP). Biological studies designed to evaluate effects of military land use on natural resources pose unique and difficult problems because collecting biological data during military activities is required. Advanced technologies that allow remote tracking and monitoring of wildlife can alleviate many of these conflicts yet provide comprehensive data.

TECHNOLOGY DEVELOPMENT

Initial Birdborne Development

The Bird-borne effort to develop a space based tracking and monitoring capability started with a study to evaluate the critical engineering paths to build a satellite received transmitter to be used on free ranging birds. Requirements for the development of the first prototype satellite transmitter were, 1) identify a space based system for transmitter development, 2) develop a PTT

under 200 grams, 3) allow for 270 days of operation, and 4) accommodate environmental, behavioral, and physiological sensors.

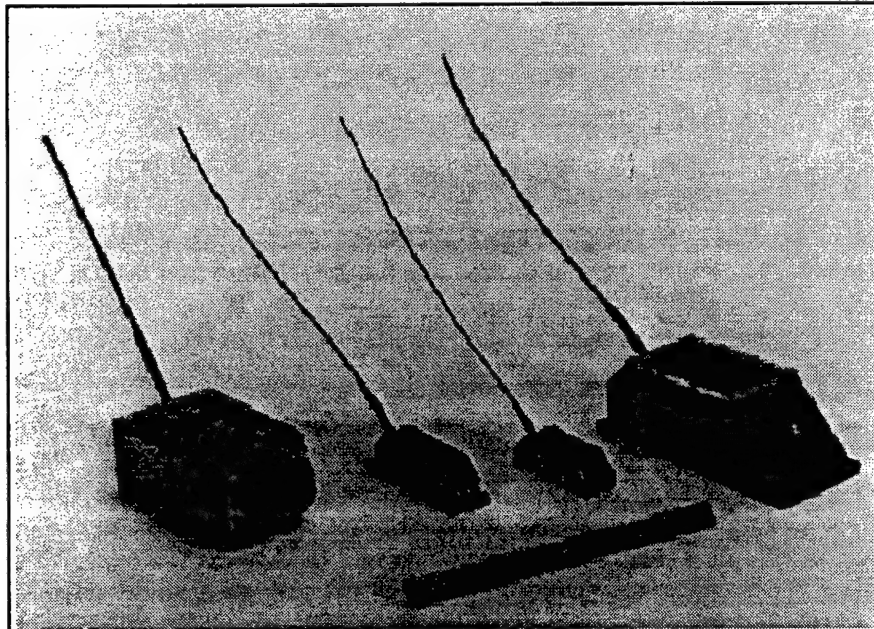


Figure 1: From Left to Right: Original 200 g Argos PTT, 30 and 20 g Nano PTTs and prototype solar powered GPS/PTT

The French operated Argos System fielded in the 1970s proved to be the basis for the development of a bird borne transmitter. The Argos system consists of receivers on the Tyros N series of National Oceanic and Atmospheric Administration (NOAA) satellites positioned in low (850 km) polar orbits. The PTTs weighed a kilogram or more and operated with

primary batteries. The PTTs are positioned based on the Doppler shift, which is dependent on a highly stable frequency transmission at 104.6 megahertz.

The initial bird borne PTT had to be relatively small (<200g), to avoid adversely affecting bird flight. System Argos required PTTs to transmit a minimum of 1.0 watt. To meet this power requirement for transmission for 270 days required 500 grams of primary batteries which exceeded by more than a factor of 2 the maximum allowed mass. Therefore, we initially met this power requirement by using a solar array with rechargeable Ni/Cad batteries. This power pack allowed for tracking duration of nearly 3 years.

During the past 15 years the electronics in the satellite transmitter have been continually miniaturized and provided new capabilities through the integration of microprocessors and mini-computers as shown in Figure 1. The newest experimental bird borne transmitters weigh 20 grams, which includes 3.5 grams of electronics, an 8 gram battery and the remainder in the container. The transmitter can interface with a variety of sensors to collect information from the environment surrounding the organism as well as collecting behavioral data. Many researchers are now applying PTTs to the study of birds [1] as well as other wide-ranging animals [2].

DIGITAL AUDIO CAPTURE AND IDENTIFICATION SYSTEM

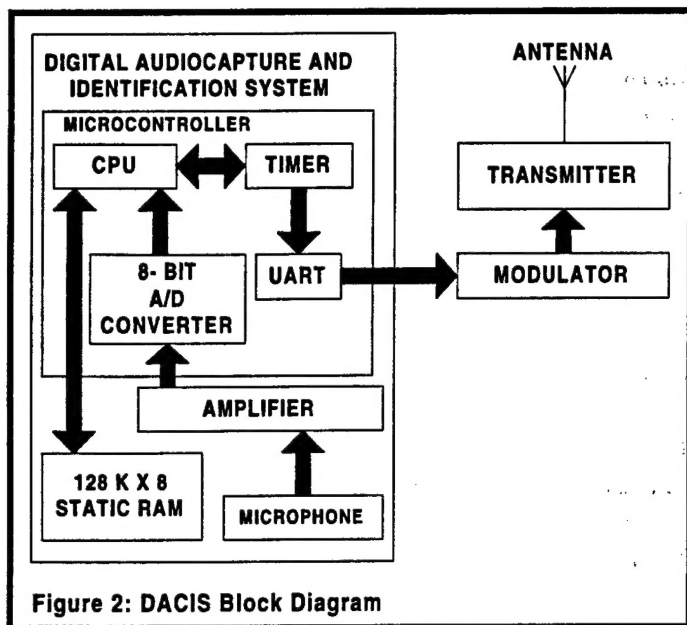
There remain, however, many questions for research and important conservation issues that need to be addressed in a timely, effective manner, and would benefit from additional development of technology. We report here on the development of a bird borne transmitter that will incorporate a Digital Audio Capture and Identification System (DACIS) that will assist in the interpretation of acoustical information to link time and location to discrete animal behaviors. The initial goals of this system are to analyze and time tag 5 distinct animal sounds from the same species or different species and transmit that data, along with the location at which the call was heard, to the Argos satellite.

Acoustic analysis of various bird calls was performed at APL by producing "waterfall" plots. These show intensity of sound as a function of frequency and time. Plots of several species appear to indicate that a system which prepares a coarse version of these plots and compares them to a known template will be able to identify different species or different calls of the same species.

System Hardware

The block diagram of the entire electronics system of the digital audio capture and control circuit is shown in Figure 2. An audio trigger, which consists of a micropower operational amplifier and voltage comparator is always on and listening to the environment. When a preset threshold is detected, the microprocessor is turned on to start the digital recording. The design of the digital audio capture circuit centers on an MC68HC811 micro controller. This device was chosen because it has several system components on a single chip and small size and weight are critical in this system's design. The subsystems of the micro controller are the internal universal asynchronous receiver transmitter (UART), the internal timer, internal Read Only Memory (RAM), Electrically Erasable Programmable Read Only Memory (EEPROM), and an 8-bit Analog to Digital (A/D) converter. The A/D converter is used to sample the amplified signal

from the electret microphone. The audio sample is then immediately stored in memory for future transmission.



The MC68HC811, which is an 8 bit micro controller, can only directly address 64K bytes of memory space. Therefore, page mode addressing is implemented using a separate output bit to control the highest address line (A16) of the memory. If more than 128K of memory is required in future versions, new circuitry will be added to control the chip select pins of the memory chip to avoid data bus clashes. There are enough spare gates and microprocessor output pins present on the existing design to increase memory to 512K bytes if desired.

The initial memory configuration, which used two 32 Kbyte memory chips was replaced with a single S-MOS systems 128K byte memory chip. The sampling rate, which can be easily changed in software, was set at 6,000 samples per second to yield reasonable quality audio playback. Initial experiments were conducted with a sampling rate of 2,700 samples per second and proved to yield marginal results for the intended system use. There is a direct trade off between

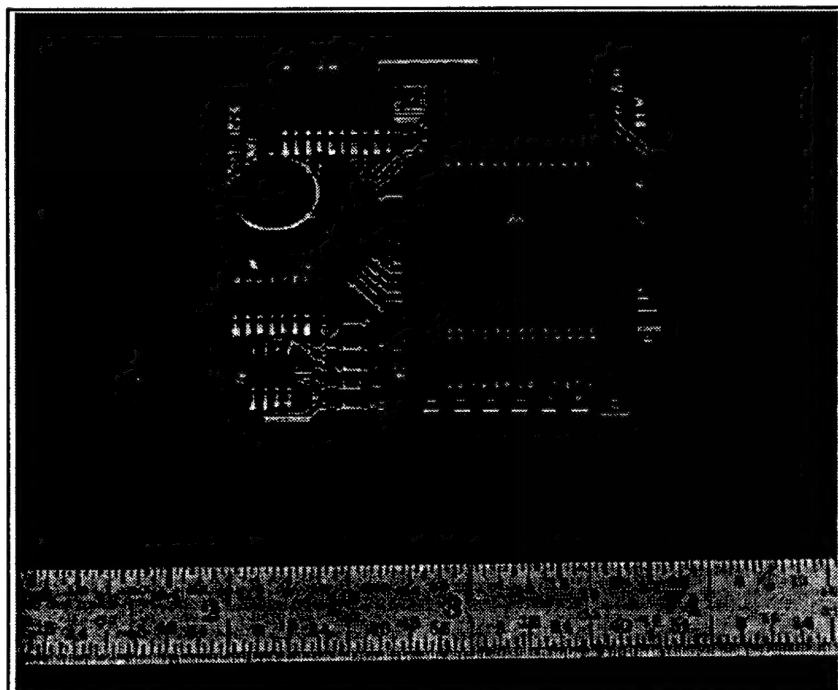


Figure 3: Prototype DACIS Board Showing Processor, RAM and other components

sampling rate and total record time. At 6,000 samples per second and using 128K (actually 131,072 bytes) less 4,096 bytes for EEPROM (and the image of EEPROM in upper memory which is inaccessible in the present implementation), the total record time available is $126,976/6,000 \approx 21.2$ seconds. Presently, only 5 seconds is recorded after each triggering event.

The microprocessor's on-chip UART is used to generate the serial data stream during transmission. The data rate is programma-

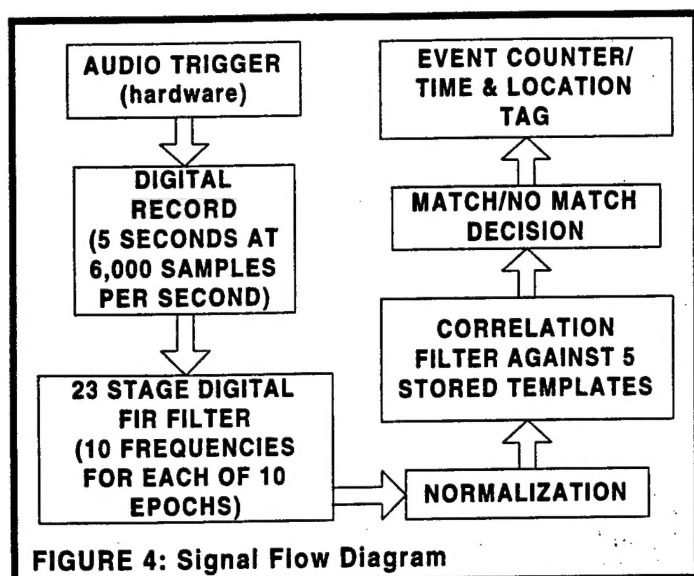
ble and has been set to 9,600 bits per second due to requirements of the prototype modulator. The data rate is not infinitely flexible in that the rate is obtained from selectable divide ratios of the microprocessor's clock. The present data format is 8 data bits per word with one start bit, one stop bit, and no parity bit. Additionally, a 74HC4024 counter is used to divide the microprocessor clock by 8 to obtain a 153.6 KHz 50% duty cycle square wave which is 16 times the data rate. This signal is required by the modulator. Lastly, during transmission, a line called transmit is brought low to activate the transmitter. This line activates the transmitter approximately 200 msec before the data starts to allow the transmitter to stabilize. This portion of the system will also serve as the control for future features such as control of a Global Positioning System (GPS) receiver. The microprocessor will then reformat the data and transmit it through the Argos satellite.

All components used in the design are available in surface mount packages. The prototype board is shown in close up in Figure 3. The board measures approximately 2" by 2", weighs about 15 grams, and has a socket for the CPU for test purposes only. A slightly smaller version has been designed without the socket for later testing

SOFTWARE

A flow chart of the signal processing is shown in Figure 4. All of the software for the system was written in 6800 series assembly language. After power up, the initial section of code simply initializes the timer counter, baud rate resistor, A/D converter and I/O pins. The audio capture portion of the software controls the internal A/D converter and stores each sample in RAM. The timing of audio sampling is controlled by interrupts from the internal timer. After all memory is filled, the program stops capturing audio and starts to process the recorded data.

The processing starts by implementing a digital 23 stage FIR (finite impulse response) filter on ten bands centered from 250 Hz to 2500 Hz in 250 Hz evenly spaced increments. The total recording of 5 seconds is divided into 10 epochs of 0.5 seconds each and the ten filters are performed on each of the ten epochs to yield 100 time/frequency parameters. This is a crude representation of the waterfall plot. The parameters are then normalized to account for differences in volume level which may disturb the pattern match.



The final processing stage is to match the 100 parameters against the 5 stored templates and do the threshold detection. These two critical stages are currently under development and testing using wolf calls. The positively identified call information, along with the time and location, will be stored for later transmission. The exact scenario by which transmission will be started has not yet been determined. The possibilities include internal timer activation, reception of a transmitted signal to the animal mounted unit or some sort of proximity activation from a

small device placed in or near the animals habitat. The last method has the advantage of allowing the receiver to be near the free ranging animal also so that the transmission distance is short. Presently, if the entire contents of the memory are sent at 9,600 baud, the total transmit time is about 132 seconds.

Global Positioning System Qualified Argos PTT

The Argos system is capable of giving locations to within $\pm 150\text{m}$ anywhere on the surface of the earth but locations obtained from tiny low power (100mW) ARGOS beacons, mounted on the backs of birds often give locations in the range of $\pm 2\text{ km}$ of the birds true locations. To achieve the highest grade Argos location at least 4 messages have to be received by the satellite over a period exceeding 420 seconds. The availability of small commercial GPS receiver modules have now made it possible to combine such a receiver with an Argos transmitter and field a package small enough to be carried by a goose size bird. By scheduling the collection of GPS locations throughout the day and storing these positions for later transmission via Argos as many as 20 GPS positions ($\pm 20\text{ m}$) can be transmitted to the user in a single Argos message.

An Argos/GPS package, under development by Microwave Telemetry Inc., incorporates a commercially available GPS receiver, a microcontroller based data logger and a Microwave Telemetry NANO PTT. The data logger controls the GPS receiver and the collection of GPS data which is dependent on power availability from the solar charged power source. The data logger then sequences data transfer to the NANO PTT at times favorable to satellite availability. The prototype unit is now undergoing laboratory testing and weighs less than 200 grams.

DISCUSSION

The technology we have described is designed for use on free ranging animals to provide data on their locations, behavior, and environment. A GPS receiver, when integrated with an Argos PTT, will provide more accurate location data that can be collected at pre-designed

times. The Argos system is dependent upon collecting frequency data on the PTT signal transmission to calculate a single time dependent location. With the use of a minicomputer integrated into the unit GPS positions can be collected according to a programmed schedule. This will increase our ability to locate free ranging organisms, and derive important facts regarding range and habitat use. With enhanced accuracy and greater numbers of locations, home range estimations, programs, and Geographic Information Systems can be used more effectively to relate animal movements to jurisdiction boundaries, habitat, and land use activity maps.

Animal sound identification data, in combination with time and location, will provide additional information relevant to natural resources. For example, the DACIS is designed to recognize animal vocalizations, thus allowing evaluation of animal behaviors and specific activities. By locating exact animal behaviors and linking them to specific habitat within the range of the organism valuable information can be collected on relationships among animals and micro-habitat components of their range. The real time and near real time components of new telemetry will allow more complete study of animal responses to a wide range of ecological variables. Time coded information on location, heading, altitude, speed, ambient temperature, humidity and other sensor data, can be displayed and analyzed relative to other geographically linked features such as geomorphology, ecological community, meteorology, and land use activities. Free ranging organisms tagged with animal track and monitor units act as a sentinel in the population. These sentinel individuals either moving alone or in herds or flocks, can reflect the activities of many organisms and enhance the biological data base dramatically.

Discussions with commercial companies have not yet identified any willing to fund the development of this system. Some have, however, indicated an interest in fabricating the unit after the proof of concept has been completed. A system which automatically trains the DACIS unit from animal sound recordings will likely be necessary to commercialize this unit. Presently, the setting of the 100 template parameters is a labor intensive task.

The use of a stand-off system to monitor and collect pertinent environmental data can be used very effectively in the evaluation of a variety of issues pertinent to many human activities. Noise effects on people as well as wildlife pose a significant problem in many areas and require a significant investment in time and money to resolve. Sensors capable of measuring environmental noise at the organism are in place and being used by the U.S. Air Force to monitor jet aircraft noise and by the Army to evaluate single event noises associated with weapons testing and training. The DOD is developing the capability to monitor the level of noise at the organism and the behavioral response to the event.

References

1. Fuller, M.R., W.S. Seegar, J.M. Marzluff, and B.A. Hoover. 1995a. Raptors, technological tools and conservation. Trans. 60th No. Am. Wildl. & Natur. Resour. Conf. pp. 131-141.
2. McConnell, B.J., C. Chambers, K.S. Nicholas, and M.A Fedak. 1992. Satellite tracking of grey seals (*Halichoerus grypus*). J. Zool., Lond. 226:271-282.